Distributed, Privacy-Aware Location Data Aggregation

1st Maja Schneider
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Dresden/Leipzig, Germany
mschneider@informatik.uni-leipzig.de

2nd Erik Buchmann
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Dresden/Leipzig, Germany
buchmann@informatik.uni-leipzig.de

3rd Erhard Rahm
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Dresden/Leipzig, Germany
rahm@informatik.uni-leipzig.de

***Abstract*—Analyzing location data from many individuals can provide valuable insights, especially when linked with private attributes like personal health information. A recent application includes identifying COVID-19 outbreaks by aggregating individuals' health data across a geographical hierarchy. However, analyzing such sensitive information can threaten the individuals' privacy, especially when honest-but-curious third parties are involved. To encourage people to share their data for such analyses, strong privacy protection and building trust in the privacy approach are crucial, requiring clear privacy parameters that can be tailored to individual needs. To address these requirements, we introduce DIPALDA, a new anonymization technique for DIstributed, Privacy-Aware Location Data Aggregation on hierarchically structured personal location data. DIPALDA leverages three privacy parameters: k-anonymity, minimum cloaking area size, and maximum re-identification probability, effectively countering re-identification and location privacy attacks. Our extensive experiments with COVID-19 propagation data demonstrate that DIPALDA achieves a suitable trade-off between utility, privacy, and explainability.**

***Index Terms*—Location privacy, Spatial k-anonymity, Privacy-aware aggregation**

I. Introduction

From a privacy perspective, it is challenging to analyze personal location data that is associated with sensitive information. This is particularly true when the evaluation is carried out by an honest-but-curious party. Such data can be used, for example, to analyze the geographical distribution of patients suffering from a certain disease, which can be helpful for healthcare planning or disease control.

Our running example is the analysis of case numbers during a pandemic: *A data analytics company aims to localize regions with high activity of an infectious disease, such as COVID-19, spreading over a country. The country is hierarchically divided into smaller areas, from "federal state" over "county", "city", "district" to "neighborhood". Persons that are showing symptoms at the time of data collection might be willing to contribute information about their infection and their area of living to help control the spread. However, no person wants to be singled out as a spreader in their neighborhood.*

A privacy-aware analysis of such data has to consider multiple issues: The location data is linked with *private information* that might reveal sensitive information about the persons concerned. The data is *distributed* among many persons who have *different attitudes* towards the level of protection required for their data. The data must be *spatially aggregated*. Neither the data collection process nor the aggregated result may reveal (a) the location, (b) the identity or (c) the sensitive attribute of a person who has contributed their data [1]. In particular, the analysis result must be robust against re-identification and location privacy attacks that can occur in this setting.

To the best of our knowledge, existing anonymization approaches are either insufficient to prevent re-identification or location privacy attacks for such use cases [2], [3], depend on a trusted third party [4]–[8], or do not allow a person to individually control their privacy parameters [2], [9], [10].

In this paper, we address these issues and introduce DIPALDA, our anonymization approach for DIstributed, Privacy-Aware Location Data Aggregation. DIPALDA's system model is visualized in Fig. 1.

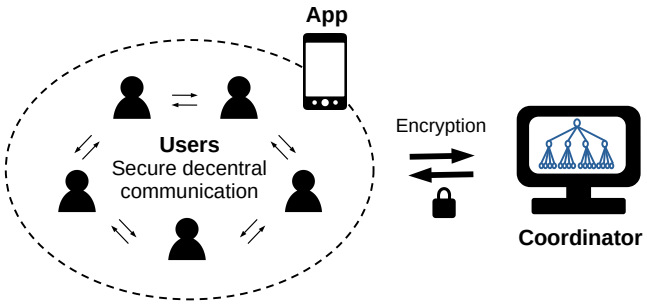


Fig. 1. DIPALDA's system model.

The aim of DIPALDA is to allow an honest-but-curious *coordinator* to collect private location information of distributed *users* and construct a hierarchical spatial index structure. An *app* orchestrates the communication between all involved parties using secure communication and encryption protocols [11], [12].

DIPALDA allows users to set individual privacy parameters k-anonymity, minimum cloaking area size, and maximum re-identification probability. While privacy approaches typically rely on Differential Privacy [13], which is considered the current standard technique, our parameters are well-explainable and their protective effectiveness against re-identification and location privacy attacks is immediately apparent.

We make the following contributions:

- We introduce DIPALDA, our new anonymization approach to construct a privacy-aware hierarchical spatial index structure from location data of many users. DIPALDA respects the users' individual privacy needs by introducing well-explainable privacy parameters, that are designed to fend off location and re-identification attacks.
- We introduce DIPALDA in the trusted setting and describe how to extend it to the honest-but-curious setting.
- We investigate with extensive experiments based on real COVID-19 propagation data how DIPALDA solves the trade-off between utility and privacy for different data distributions. Our evaluation shows that DIPALDA can balance utility and privacy for a real use case.

The rest of the paper is organized as follows: Section II reviews related work. Section III introduces the problem. Section IV describes DIPALDA and its privacy properties, followed by an evaluation in Section V. Section VI explains how other spatial hierarchies can be used. Section VII concludes.

II. Related work

A. Privacy-preserving location data aggregation

First approaches to protecting sensitive tabular user data are based on k-anonymity [14], [15]. This concept was introduced in the context of private data release from relational databases where a combination of quasi-identifying attributes can threaten to re-identify a user and leak their sensitive attribute. In order to achieve k-anonymity, anonymity groups of at least k users are formed, so that group members are indistinguishable from each other with respect to their quasi-identifiers. To specifically address the protection of location data, k-anonymity was extended to spatial k-anonymity (SKA) [6]. Following the concept of anonymity groups, SKA requires a user's location to be spatially and temporally indistinguishable from the locations of $k-1$ other users.

K-anonymity is typically achieved by generalization and suppression techniques. Because the generalization of data leads to data loss, privacy-preserving methods aim to maintain high data quality while protecting privacy. Several works suggest that hierarchical index structures are well suited to perform efficient privacy-preserving generalization over multiple quasi-identifying attributes [16]–[18]. Because spatial index structures are an obvious choice to partition geographical space, they have been widely used by spatial anonymization approaches to implement SKA, particularly in the context of Location-based Services (LBS) [4]–[7], [19]–[22]. In the LBS scenario, a user sends a spatial query to an untrusted server, e.g., finding nearby points of interest. Privacy-preserving techniques aim to find a suitable cloaking region for the user that contains at least $k-1$ other users to be sent to the server instead of the user's exact location. This limits the re-identification likelihood of the user to at most $1/k$. Due to its good explainability, k-anonymity is still widely adopted in practice. The sole use of spatial k-anonymity is, however, not sufficient to prevent location homogeneity and background knowledge attacks. Approaches relying on k-anonymity thus require further protective measures to fend off such attacks.

To overcome the limitations of previous approaches, Differential Privacy (DP) [13] was introduced. DP is a formally verifiable privacy guarantee that limits the influence of a data point on an aggregation result. This is typically achieved by adding noise to the individual or aggregated user data before it is further processed by an untrusted third party [8], [23]–[25]. The level of privacy is controlled by a privacy budget ϵ which influences the amount of noise to be applied. To achieve a suitable balance between utility and privacy of the analysis result, the calibration of the required noise is a topic of research [24]–[29].

While DP provides a mathematical formulation to limit privacy leaks, its main drawback is that the privacy budget ϵ is an abstract value which can be hard to understand without expert knowledge [30]. Its protective capabilities against certain attacks, such as re-identification and location privacy attacks, are not directly visible [31]. Furthermore, calibrating the privacy budget to provide sufficient protection against such attacks while maintaining utility often depends on the use case and data distribution [32]. Because in our use case the data owners should be able to choose their own level of privacy, techniques are needed that are both easy to explain and provide effective protection against the aforementioned attacks.

B. Secure communication and encryption

To achieve SKA or DP, some privacy concepts require a trusted central server to provide the necessary sanitization of the data before it can be shared with an untrusted party [4]–[7], [22]. However, the use of a trusted server itself poses risks to privacy. To eliminate this dependency, some approaches rely instead on a decentralized anonymization process within peer groups [19]–[21].

To further limit assumptions on the required trust model, secure communication protocols and encryption techniques are employed, such as Secure Multiparty Computation (SMPC) [12] and Homomorphic Encryption (HE) [11]. SMPC allows multiple parties to jointly calculate a result based on their private inputs without any party learning anything other than their own input and the result. SMPC is often combined with HE, which enables secure communication by encrypting the data while still allowing calculations on the encrypted data.

Related work exists in spatial crowdsourcing which requires the assignment of task locations to worker locations in a private manner. Some solutions make use of spatial structures, such as grids or trees, in combination with encryption techniques, to encode locations and privately retrieve distances between workers and tasks [2], [3]. The authors of [2] propose a grid-based location privacy framework but their approach does not account for individual privacy requirements of users. A decentralized solution is presented in [3] based on blockchain technology and HE. The approach allows for individual privacy settings of users but does not provide provable privacy guarantees for the released information against re-identification.

III. Problem description

Our goal is to construct a hierarchical spatial index structure H from the private locations of a set of users $u \in U$ that have a certain sensitive attribute. The aggregation result must not reveal more private information about a user than the user is willing to share, including their identity, location and sensitive attribute. In particular, the result must be robust against a set of re-identification and location privacy attacks. In this section, we describe the single building blocks of this problem and illustrate them using our running example.

A. User data

We assume a set of users U where each user $u \in U$ has a certain binary sensitive information $s_u \in \{true|false\}$. At time of data collection, users with a positive sensitive attribute contribute their location l_u, which is a spatial point, e.g., described as a pair of latitude and longitude. Our focus is thus the aggregation of location snapshots, which means that we do not consider repeated location updates of a mobile user. A user can set three privacy parameters which are explained in more detail in Section III-D. *In our running example the users U refer to the population in a geographical region. The location refers to a user's home address and their sensitive attribute indicates whether they show COVID-19 symptoms.*

B. Spatial index structure

While DIPALDA is applicable to any data-independent hierarchical data structure, we introduce our approach based on a Region Quadtree [33]. This data structure $H = (V, M, h_{max})$ recursively divides space into $M = 4$ equal subareas until a maximum depth h_{max} is reached. H contains a set of vertices V. Each vertex $v = (r_v, U_v, n_v)$ describes a geographical bounding box r_v and contains a subset of users U_v with $s_u = true$ and $l_u \in r_v$. The aggregation result of a vertex is its number of users $|U_v|$. A vertex furthermore contains the number of inhabitants n_v, where $|U_v| \leq n_v$. *We regionally aggregate users showing COVID-19 symptoms with a Region Quadtree in order to identify disease spreading hotspots. Fig. 2 illustrates this. With increasing hierarchy levels in the tree, from the root to the leaf nodes, the respective subareas become smaller and thus enable more detailed analyses. At the same time, data is becoming increasingly scarce, increasing the risk of the privacy of the analyzed users being violated.*

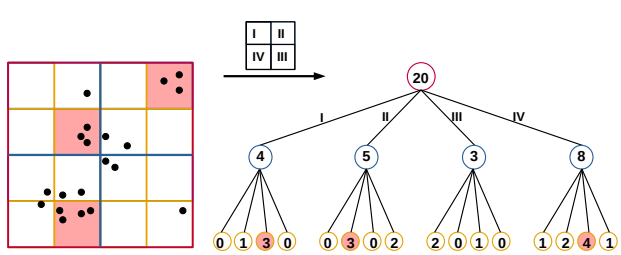


Fig. 2. Aggregation of locations into a Region Quadtree.

C. Privacy threats

The aggregation of such user data bears the risk that a user's identity, location or sensitive attribute is exposed by an attack.

Restricted space identification attack [6]: This attack exploits the fact that a user provided their location information at too high a spatial resolution which can cause re-identification. *If a user is contained in a vertex on a deep hierarchy level, the corresponding region can be so small that it only includes premises that can be associated with the user, e.g., the house where the user lives. In this case, the user can be re-identified.*

Location homogeneity attack [1]: Some privacy approaches based on spatial k-anonymity [6] attempt to prevent re-identification by reporting a cloaking area instead of a user's exact location so that at least k users' locations are indistinguishable. However, if these locations are geographically close to each other, any user's true location is known with only small error. *The home location of a user can be narrowed down to the region of the vertex in which the user is contained. Thereby, it is irrelevant how many users are contained in that vertex in total.*

Advanced location homogeneity attack [1]: An attacker can increase their certainty about a user's location by using external information, such as maps, to exclude unlikely regions (e.g. lakes). *If a user is located in a vertex whose region is 90 % covered by water, the user's home location can be narrowed down to 10 % of the area.*

Background knowledge attack [34]: An attacker can use background knowledge to increase their certainty about whether a certain individual possesses the sensitive attribute. *It can happen that a user is contained in a vertex that covers only one single building with at least k inhabitants. K-anonymity makes the user indistinguishable from $k-1$ other users. If, however, the aggregation result, i.e., the number of sensitive users in that vertex, approaches the number of inhabitants living in this region, an attacker can simply look up the residents of that building and infer for each of them a high likelihood to be COVID-19 positive.*

D. Privacy protection

In order to fend off re-identification and location privacy attacks mentioned in Section III-C, DIPALDA allows each user u to set three privacy parameters: a minimum cloaking area size that is translated into a hierarchy level h_u, a spatial k-anonymity parameter k_u, and a maximum re-identification probability p_u.

The **minimum cloaking area size** [6] describes the maximum spatial resolution with which a user is willing to share their location. With DIPALDA, this information is represented as the deepest hierarchy level h_u in H that user u can be present in. The hierarchy level can be intuitively interpreted as "city", "district", "neighborhood", etc. h_u is calculated so that the size of a subarea on this level matches or exceeds the cloaking area size. Furthermore, census statistics are taken into account to ensure that the population distribution does not allow (advanced) location homogeneity attacks. From the user's location, first the vertex v on level h_u is derived for

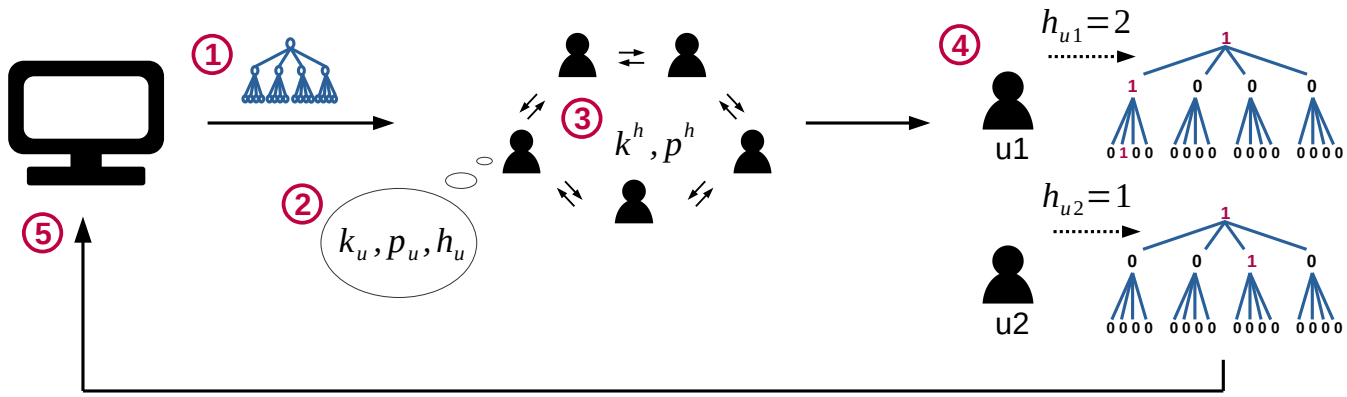


Fig. 3. The processing steps of DIPALDA.

which $l_u \in r_v$. The hierarchy level h_u is iteratively decreased as long as the population count n_v in vertex v is concentrated in one of the M child vertices of v on hierarchy level $h_u + 1$.

With the **spatial k-anonymity** parameter [6] k_u, the user requires a lower bound for the number of users that are included in the same area and at the same hierarchy level of H. k_u makes sure that the user is spatially indistinguishable from at least $k_u - 1$ other users and thus counters re-identification due to a restricted space identification attack.

With the **re-identification probability** p_u a user defines by how much at most they accept to be re-identifiable as someone with a positive sensitive attribute among all inhabitants of an area. The user will thus be only present in vertices with a smaller or equal re-identification likelihood $p_v = |U_v|/n_v$ where $|U_v|$ is the number of users that are contributing their data to a vertex v and n_v is the population size from census associated with the vertex. This parameter prevents that an adversary with background knowledge about population sizes can learn with high certainty that a user is a spreader, because the user resides in a neighborhood where almost every inhabitant is infected. If a user u with a positive sensitive attribute $s_u = true$ requires a small p_u this means that there needs to be a comparatively large number of users with $s_u = false$, allowing the user plausible deniability and countering attacks with background knowledge on population sizes. Because this parameter is a probability, its interpretation is intuitive and does not require users to do any calculations.

IV. Our DIPALDA Anonymization Approach

In this section, we describe DIPALDA, our approach for a DIstributed, Privacy-Aware Location Data Aggregation. The goal of DIPALDA is to allow a central party, the coordinator, to construct a spatial index structure from the locations of many users while preventing re-identification and location privacy attacks on the aggregation result. On the user side an individually configured app takes care of implementing the required communication and protocols. We introduce our approach on the example of a Region Quadtree.

To simplify our presentation, we first explain our anonymization approach using a trusted coordinator and trusted users. Second, we describe in Section IV-C how DIPALDA can be implemented in the honest-but-curious setting.

A. Processing steps in the trusted setting

The outline of DIPALDA is shown in Fig. 3. We assume a coordinator C to manage the index structure H. C initiates the data collection process by publishing required information on the data structure via the users' apps. The users jointly calculate shared privacy parameters for the index structure and anonymize their location accordingly to align with their individual privacy needs. C ensures that the shared privacy parameters are met in the final aggregation result. In the following, we explain these steps in detail.

(1) Publication of data structure and its parameters: Initially, C decides which data structure H is to be used and informs all users via the app of the parameters required to create it. For a Region Quadtree, these parameters include the degree M, the maximum tree depth h_{max}, and the bounding box r_{root} of the geographical region to analyze. C furthermore publishes the population size n_v for each vertex $v \in V$ based on census information.

(2) Setting of individual privacy parameters: The users then configure their app with their individual privacy parameters: the minimum number k_u of users that must be present in a vertex, the maximum re-identification probability p_u, and the minimum cloaking area size. The app translates the cloaking area size into a hierarchy level h_u as described in Section III.

(3) Calculation of shared privacy parameters: The next step is to define the privacy guarantees that should apply to the final aggregation result. To this end, shared privacy parameters k^h and p^h are agreed for each hierarchy level $h \leq h_{max}$. The challenge here is to find suitable parameters that make it possible to include a maximum number of users in the aggregation result while respecting their individual privacy parameters. While multiple approaches exist to identify suitable parameters[1], we propose the following approach:

[1]For example, if C has background knowledge that allows it to properly estimate the distribution of k^h and p^h, C can define default parameters for all levels and the decentralized calculation in step (3) can be skipped.

DIPALDA first calculates an estimate of the sensitive user count $|U_v|$ in a vertex by dividing the sensitive user count $|U_{root}|$ in the root vertex by the number M^h of vertices in the respective level h. Furthermore, a re-identification probability p_v is estimated for each vertex v by dividing $|U_v|$ by the population size n_v. Privacy parameters are then set as

$$k^h = \begin{cases} (1-\delta) * |U_v|, & \text{if } |U_v| < \max_u k_u \\ \max_u k_u, & \text{otherwise} \end{cases} \quad (1)$$

and

$$p^h = \begin{cases} (1+\delta) * p_v, & \text{if } p_v > \min_u p_u \\ \min_u p_u, & \text{otherwise} \end{cases} \quad (2)$$

with $\delta << 1$ being a small number. These calculations aim to find a balance between including as many users as possible by allowing strict privacy requirements and excluding users with too strict privacy requirements that are unlikely to be met. The parameter δ controls this trade-off.

The shared parameters are broadcasted among the users and the coordinator. Note that this approach requires the sensitive user count $|U_{root}|$ in the root to be known, which limits the privacy guarantees for this level.

(4) Location encoding: Each user's app finds the level h for which the user's individual privacy parameters are not in line with the shared parameters, i.e. $k_u > k^h$ or $p_u < p^h$. Then, the app updates the user's individual maximum hierarchy level to $h_u^{updated} = \min(h_u, \max(0, h-1))$ accordingly. For each vertex $v \in V$ the user's app creates an encoding x_v of the user's location l_u. This value is $x_v = 1$ if the user's location l_u is within the region r_v associated with the vertex, otherwise $x_v = 0$. For $h > h_u$ the encoded value is $x_v = 0$ even if the user's location is inside the spatial boundaries of r_v.

(5) Aggregation: The encoded values are sent to the coordinator who calculates the aggregated sum for each vertex. C makes sure that the aggregation result in each vertex complies with the shared privacy parameters that have been decided in step (3). To this end, C calculates a conditional function $f_v = c_v * |U_v|$ over the sum $|U_v|$ of the user inputs where

$$c_v = \begin{cases} 1, & \text{if } k^h \leq |U_v| \leq p^h * n_v \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Finally, C constructs the tree H from the acquired information.

B. Privacy in the trusted setting

In constructing the spatial index structure H, DIPALDA safeguards that no private information about any user can be leaked from the aggregation result due to re-identification and location privacy attacks, mentioned in Section III-C. This includes the protection of each user's location information, their identity, and sensitive attribute. DIPALDA enforces this by a) allowing each user to set individual privacy parameters that are specifically designed to prevent such attacks, b) making sure in step (4) that a user's location is only encoded for those hierarchy levels where their individual privacy parameters are in line with the shared privacy parameters of the data structure, and by c) making sure in step (5) that the aggregation result in each vertex of the data structure is only released, if the shared privacy parameters for this vertex are fulfilled.

C. From trusted to honest-but-curious

After presenting our anonymization approach in the presence of trusted parties, we explain how DIPALDA can be extended to provide privacy in the honest-but-curious setting. In this case, all parties involved, including the coordinator, users and external adversaries, follow the protocol, but try to obtain as much information as possible about the users. We further assume, that parties do not collude.

In the presence of honest-but-curious external adversaries there is a risk of Sybil attacks [35]. In this attack, an adversary registers for the app with one or more fake identities which endangers SKA in the aggregation result due to an increased user count. To prevent this, the users can verify their identity with a certificate before the data collection takes place, e.g., using self-certified pseudonyms [36].

Further additional protective measures are required to ensure that no sensitive user data is revealed during the data aggregation process. In particular, we require (i) that the information that is distributed to the users' apps in step (1) is correct, and (ii) no sensitive user information is leaked during communication in steps (1), (3) or (5).

To fulfill requirement (i), app signatures can be used, making sure that the app can not be manipulated, and transmitted messages are not tampered with. In order to conceal the users' identities during all communication steps for requirement (ii), the app can create random user pseudonyms and use anonymous routing protocols. To conceal the users' individual privacy parameters, the calculation of shared privacy parameters in step (3) can be carried out with secure comparison protocols, e.g., [37].

In step (5), we need to protect the users' encoded locations before being sent to the coordinator, as well as their aggregated value in case it does not fulfill the shared privacy requirements. Therefore, we propose to encrypt the encoded user locations using a Fully Homomorphic Encryption (FHE) scheme that allows comparison of encrypted values, e.g., BGV [38]. A distributed key generation scheme can be used to avoid that a single party has access to the users' encryption keys, e.g., Threshold FHE as described in [39]. The proposed process is depicted in Fig. 4.

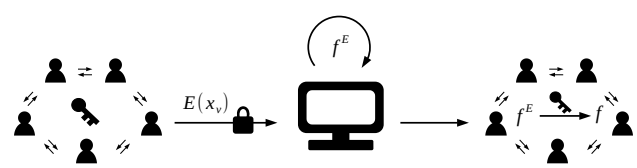


Fig. 4. Aggregation of encrypted locations.

First, each user creates an individual secret and public key and sends the public key to the other users. Each user then calculates the sum of the received public keys and uses it to encrypt their location. The encryption key is thus secret

shared [40] among all users. After receiving the encrypted data from the users, C calculates the conditional sum for each vertex v and obtains the encrypted result f_v^E. C broadcasts the result back to the users who each partially decrypt it with their individual secret key. The users jointly calculate the sum over the partial results using a secure aggregation scheme [12] and obtain f_v. Finally, C receives the sum f_v for each vertex v from the users and can construct the tree.

V. Evaluation

In this section, we evaluate our anonymization approach DIPALDA *in the trusted setting*. We are thus specifically interested in the utility of our approach concerning different privacy parameters. To this end, we study the use case of discovering regions of high COVID-19 activity based on realistic data obtained from real census information and COVID-19 statistics. Note, that an analysis of performance and complexity in the honest-but-curious setting is out of the scope of our paper.

A. Data set

We simulate a set of users U by sampling random locations according to real population distributions for Europe [41]. We then assign the sensitive attribute (being COVID-19 positive) to a set of randomly selected users to reflect actual COVID-19 infection rates in a given subregion during a period of medium to high pandemic activity. Subregions refer to districts and independent cities [42].

We study the impact of different data distributions on the utility-privacy trade-off of DIPALDA by selecting two urban, rural and mixed regions each, characterized by varying population densities. A rural area is defined by a number of less than 300 inhabitants per km^2, and an urban area by more than 1.500 inhabitants per km^2. To investigate the impact of the COVID-19 infection rate, we select regions with varying infection rates. Table I summarizes the population numbers and COVID-19 statistics of the analyzed regions.

B. Experimental setup

We investigate the influence of different attitudes of a user towards their privacy requirements, ranging from very relaxed to very restrictive. To this end, we define multiple value ranges for each privacy parameter that reflect increasingly stringent privacy requirements: $k = [0, 5]$, $[5, 20]$, $[20, 40]$, $[40, 60]$, $[60, 100]$, $p = [0.5, 1]$, $[0.1, 0.5]$, $[0.05, 0.1]$, $[0.03, 0.05]$, $[0, 0.01]$, $h = [7, 7]$, $[4, 7]$. A user's individual privacy parameters are randomly drawn from a uniform distribution over a given value range.

We calculate anonymized Quadtrees H using DIPALDA for each region and privacy parameter combination. The shared privacy parameters k^h and p^h, that apply to each vertex in a hierarchy level h of H, are estimated as described in Section IV using $\delta = 5\%$. As a baseline we use the ground truth, which are complete Quadtrees constructed without any privacy constraints. [2] For both approaches we fix the maximum hierarchy level at $h_{max} = 5$, which corresponds to a region size between $1.3\,\mathrm{km}^2$ (Milan) and $7.8\,\mathrm{km}^2$ (Florence area).

[2]We assume for our evaluation that all users of the population, who are COVID-19 positive, participate in the analysis and thus the aggregation results in the baseline reflect the true distribution of COVID-19 cases.

TABLE I
Regions.

Region	Area type	Size in km^2	Population per km^2	COVID-19 patients per 100k
Berlin (DE)	Urban	880	4,111	3,582
Milan (IT)	Urban	1,604	2,006	3,180
Cologne area (DE)	Mixed	26,177	547	2,314
Florence area (IT)	Mixed	23,177	151	4,496
Lüneburg area (DE)	Rural	15,551	114	1,854
Jylland area (DK)	Rural	13,234	99	9,274

Experiments were carried out on a AMD(R) EPYC(R) 7551P @ 2.0GHz - Turbo 3.0GHz CPU, 512 GByte memory on Rocky Linux 8 using Python 3.10 and a PostgreSQL 16.0 database with PostGIS 3.4.0 extension. [3]

C. Evaluation metrics

Following [27], we evaluate the utility loss caused by anonymization in DIPALDA with the Relative Error (RE). The Relative Error is calculated as the normalized error between the anonymized user counts $|\hat{U}_v|$ and the true user counts $|U_v|$, calculated over a set of vertices $v \in V$ in the tree H:

$$RE(V) = \sum_{v \in V} \frac{||\hat{U}_v| - |U_v||}{|U_v|} \tag{4}$$

The Relative Error serves as a lower bound for the utility loss. However, to understand DIPALDA's performance on real world applications, we additionally consider a classification task on H that answers the question: *Can we identify regions with high COVID-19 activity that indicate disease spreading activity?* We thus evaluate for each vertex v whether its COVID-19 infection rate I_v surpasses a certain threshold I^{thr}, indicating high COVID-19 activity:

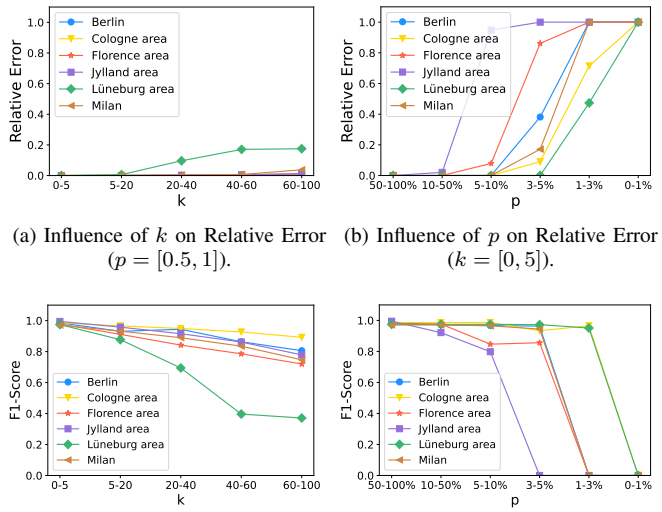
$$high\text{-}activity(v) = \begin{cases} 1, & \text{if } I_v \geq I^{thr} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where the infection rate I_v is the number of reported COVID-19 patients $|U_v|$ normalized by the census population count n_v in the corresponding region. The anonymized infection rate is calculated accordingly over the number of COVID-19 patients $|\hat{U}_v|$ in the anonymized tree. Based on the classification results in the true and anonymized tree we calculate the F1-Score over all vertices of the complete tree. The F1-Score is calculated as the harmonic mean of Precision and Recall.

D. Impact of the privacy parameters

In this subsection, we describe how DIPALDA's privacy parameters k, p and h influence the utility of a spatial index H. The influence of a user's maximum hierarchy level h is intuitive, as it limits the hierarchy levels that the user will be

[3]The code is available at https://github.com/majaschneider/DIPALDA.



(a) Influence of k on Relative Error ($p = [0.5, 1]$).

(b) Influence of p on Relative Error ($k = [0, 5]$).

(c) Influence of k on F1-Score ($p = [0.5, 1]$).

(d) Influence of p on F1-Score ($k = [0, 5]$).

Fig. 5. Influence of k and p on Relative Error and on F1-Score ($h = 7$), calculated over all vertices in the tree.

included in. For the two tested value ranges of h we achieve similar results and therefore only show results for $h = 7$.

To understand the importance of k and p for the utility of H, we fix k, respectively p, at the most relaxed value and observe the Relative Error and the F1-Score of the disease spread classification task while varying the other privacy parameter. The results are shown in Fig. 5. For all tested k-value ranges and for almost all regions the Relative Error stays below 4%, indicating a small influence of k onto the error. An exception is Lüneburg area with a Relative Error of up to approximately 18%. The influence of k on utility is more evident, reducing the F1-Score from almost 100% to 72-89% in almost all regions, and to 37% in Lüneburg area.

With tighter privacy restrictions for the re-identification probability p, the Relative Error is increasingly noticeable, becoming maximal for all regions at the most restrictive value of p. The experiments show for the tested regions that a higher COVID-19 infection rate leads to a faster increase of the Relative Error with more restrictive values of p. This observation can be explained by a correlation between the Relative Error and the infection rate I_i in a vertex v_i. Consider that all users in the vertex set $p < I_i$, then each of these users will be excluded from the vertex's aggregation result because their privacy requirement can not be met. The results are in line with this observation, as p values smaller than a region's infection rate lead to a Relative Error of approximately 100%. The parameter p has similar impact on the utility, with F1-Score dropping from approximately 80% to 0% when the Relative Error is surpassing 95%. The results indicate that the re-identification probability p has higher influence on the utility of a spatial index than k.

E. Impact of the population density

It is to be expected that a high Relative Error is caused by either a too high ratio of sensitive users, surpassing p, or a too low number of sensitive users, falling below k. These cases correlate with a high infection rate and a low population size. Our experiments confirm this assumption. The Relative Error for all tested privacy settings is shown in Fig. 6. In regions with a high population density (e.g., Berlin and Milan) or with low infection rates (e.g., Cologne and Lüneburg area) the Relative Error is comparatively low, given that p is not too restrictive (above 3%). Lower values of $p \leq 3\%$ lead to a high error in all regions. The highest errors occur in regions with comparatively high infection rates and low population density, such as Jylland area and Florence area.

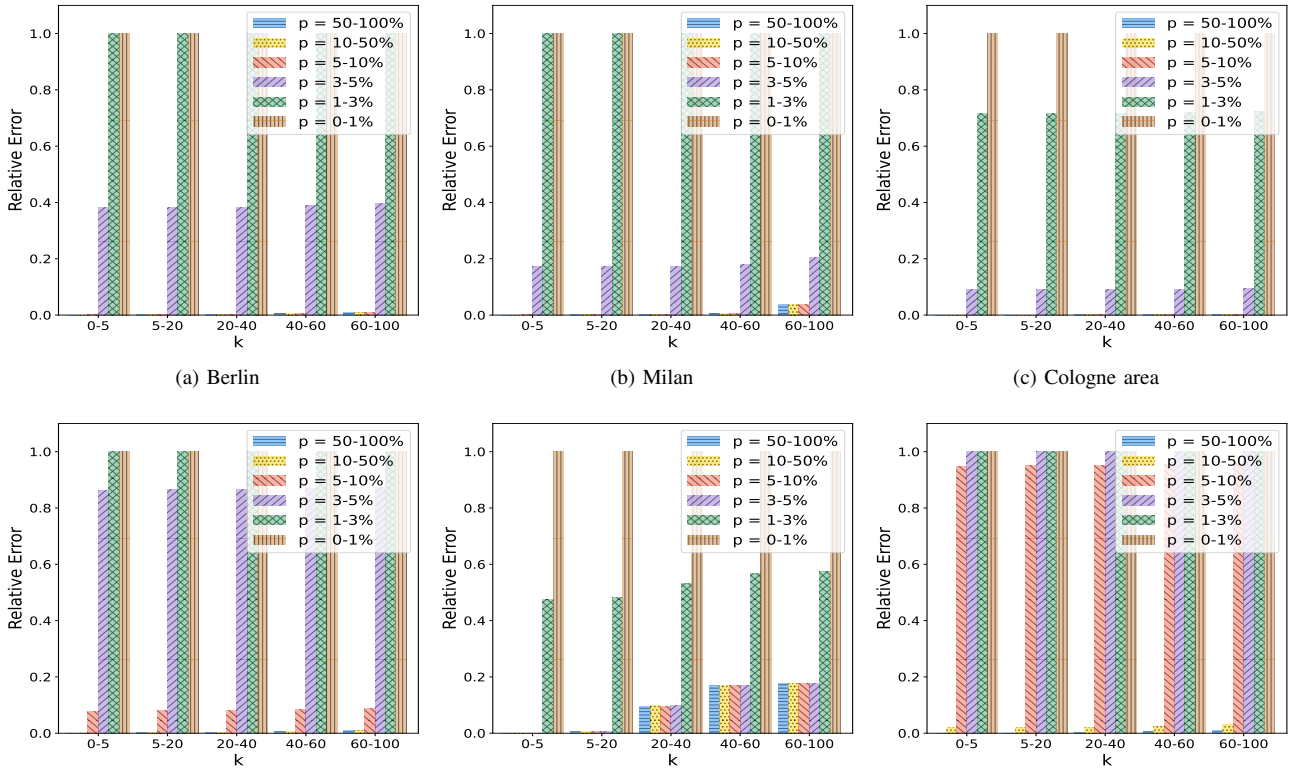
The distribution of false negatives for the classification task in Fig. 7 visually confirms our finding. False negatives represent vertices where a high COVID-19 activity was not detected. For better visibility privacy settings are chosen that achieve a low F1-Score. The figures show that false negatives appear mainly in the less populated outskirts of the cities, or in rural subregions.

F. Utility-Privacy Trade-Off

We evaluate the usefulness of DIPALDA for a realistic application, which is to detect regions with high COVID-19 activity. Fig. 8 shows the F1-Score for this task. As expected, in all regions, a downward trend in utility can be observed when privacy parameters p and k become more restrictive. However, as can be seen in Fig. 9, even a high Relative Error of up to 80% achieves mostly an F1-Score of over 60%. While the Relative Error can rise quickly if p is not chosen carefully, it stays below 4% in non-rural areas (and below 18% in rural areas) for a comparatively strict re-identification likelihood of $p \geq 10\%$. For such values of $p \geq 10\%$ the F1-Score ranges from 72% to 99% with a median of 91% in non-rural areas, and from 61% to 99% with a median of 89% when p is further restricted to $5\% \leq p \leq 10\%$.

Because Jylland area and Florence area both have a high number of subregions with low population density and high infection rates, they incur the biggest utility loss. In regions with such preconditions, accurate results can only be achieved with less restrictive privacy parameters. Recommendable values are $k \leq 40$ and $p \geq 10\%$ to achieve an F1-Score of more than 68%. For urban or mixed regions, where the population density is overall higher, a $p \geq 5\%$ achieves an F1-Score of at least 61%, and $p \geq 10\%$ an F1-Score of at least 72%.

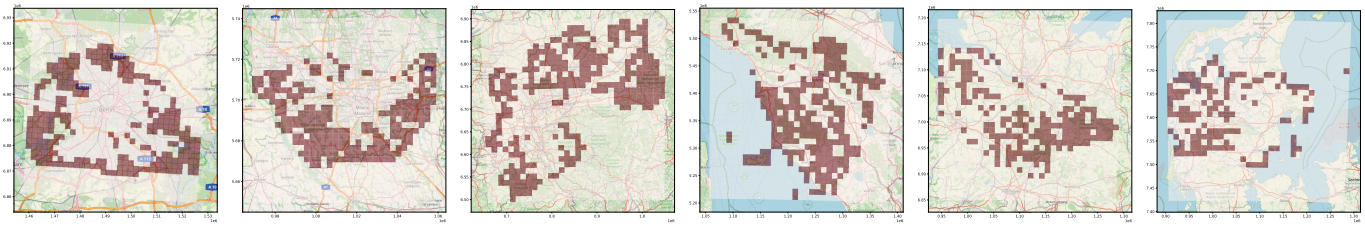
To get an intuition at what level of detail an anonymization with DIPALDA impairs the utility of the analysis too much, we observe the Relative Error and F1-Score with increasing hierarchy levels. Fig. 10 shows this data for a tree where p is fixed to $[0.1, 0.5]$. In most regions an analysis up to hierarchy level $h \leq 4$ achieves a sufficient F1-Score of at least 80% (except for Jylland area with an F1-Score of 25%). The biggest utility drop appears in hierarchy level $h = 5$, but this depends strongly on the region.



(a) Berlin (b) Milan (c) Cologne area

(d) Florence area (e) Lüneburg area (f) Jylland area

Fig. 6. Relative Error ($h = 7$).



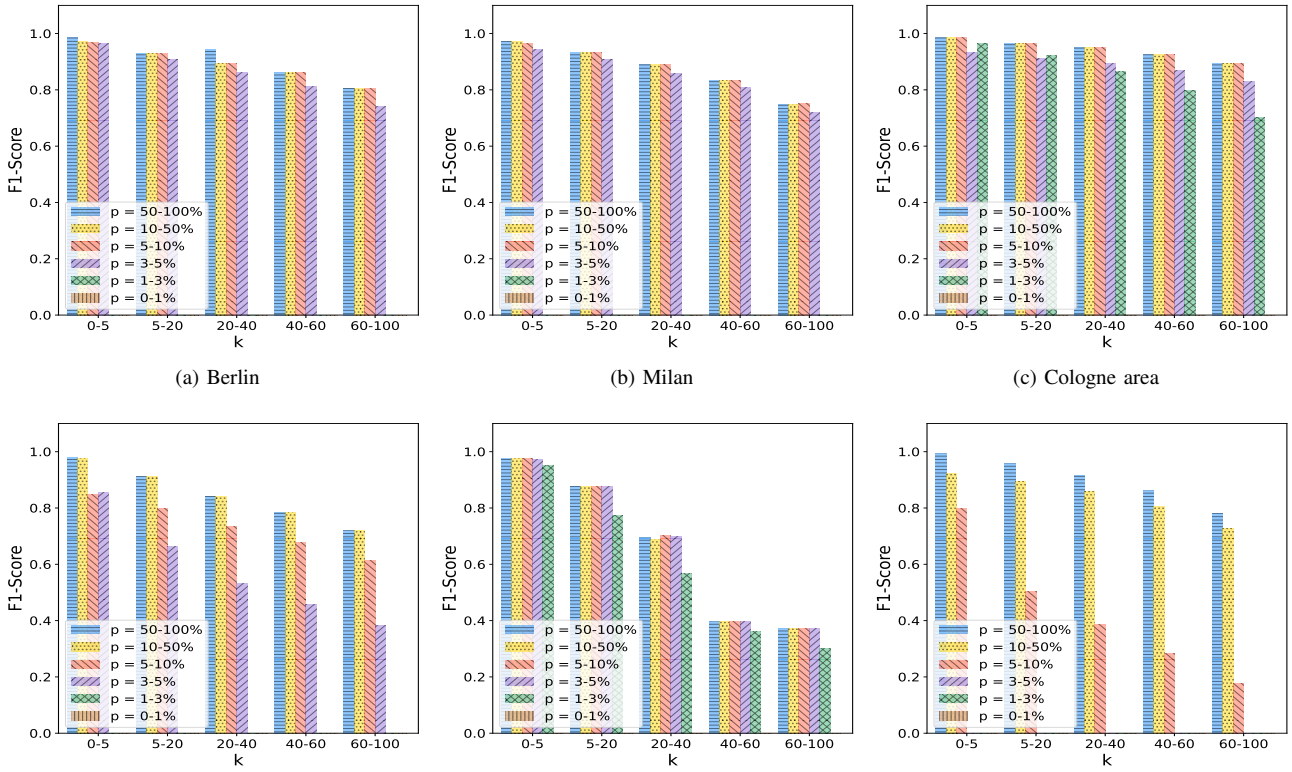
(a) Berlin ($p = [0.03, 0.05]$, $k = [60, 100]$) (b) Milan ($p = [0.03, 0.05]$, $k = [60, 100]$) (c) Cologne area ($p = [0.01, 0.03]$, $k = [60, 100]$) (d) Florence area ($p = [0.05, 0.1]$, $k = [40, 60]$) (e) Lüneburg area ($p = [0.01, 0.03]$, $k = [5, 20]$) (f) Jylland area ($p = [0.05, 0.1]$, $k = [0, 5]$)

Fig. 7. Distribution of false negatives on level $h = 5$ of a tree anonymized with DIPALDA.

VI. Discussion

To simplify the representation, we have defined DIPALDA on the basis of a Region Quadtree which describes square areas of equal size per hierarchy level. However, DIPALDA can be applied to any data-independent hierarchical index with a predefined structure that forms non-overlapping areas per hierarchy level. This enables further use cases that rely on spatial hierarchies with a different semantic meaning. For example, DIPALDA can be used on a hierarchy of census or administrative boundaries, e.g., to privately analyze the employment status in a population. Another example is the analysis of sales activities in a company based on a custom internal hierarchy that ranges from larger sales areas down to single customers. Note, that the use of such different hierarchies can lead to a different interpretation of the population size n_v in a vertex, e.g., in the latter case as the number of potential customers in a region.

In order for such applications to be feasible in the real world, it is essential to allow the data owners to control their own privacy settings. This promotes trust and acceptance of the privacy approach. For this reason, our goal was not only to consider the trade-off between utility and privacy, but also explainability. In contrast to Differential Privacy, DIPALDA privacy parameters lead to more explainable privacy guarantees that prevent re-identification and location privacy attacks. The link between our privacy parameters and the protection from such attacks is with DIPALDA directly visible.



(a) Berlin (b) Milan (c) Cologne area

(d) Florence area (e) Lüneburg area (f) Jylland area

Fig. 8. F1-Score ($h = 7$).

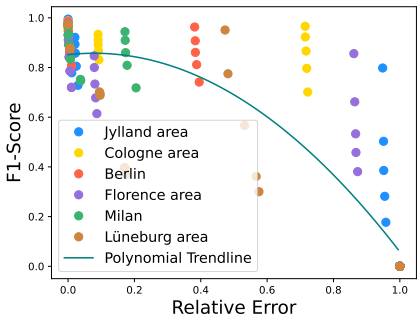
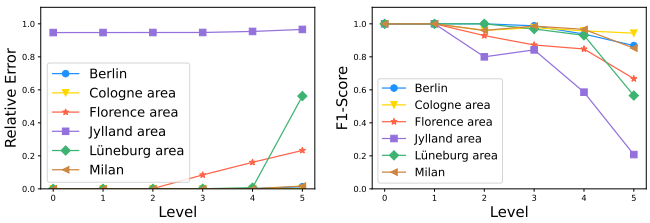


Fig. 9. Relative Error versus F1-Score.



(a) Relative Error (b) F1-Score

Fig. 10. Relative Error and F1-Score per hierarchy level ($h = 7, p = [0.05, 0.1], k = [20, 40]$).

VII. Conclusion

In this paper, we studied the aggregation of private locations from many distributed users that possess a certain sensitive attribute, such as a COVID-19 infection. We introduced DIPALDA, our new anonymization approach, that solves this task without requiring a trusted central party. DIPALDA allows each user to specify three well-explainable privacy parameters, which are used to generate an aggregation result that is secure against re-identification and location privacy attacks. We have tested DIPALDA with COVID-19 propagation data. Our experiments show that DIPALDA can achieve a suitable trade-off between utility and privacy. In future work, we plan to practically evaluate different attacks in a comparative study with Differential Privacy approaches. We also plan to extend our evaluation to examine the performance of our approach in the honest-but-curious setting.

Acknowledgment

The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the program Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI.

The authors acknowledge the use of the GISCO statistical unit dataset by European Commission – Eurostat/GISCO [42] © EuroGeographics for the administrative boundaries.

References

[1] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, 2014.

[2] D. Yuan, Q. Li, G. Li, Q. Wang, and K. Ren, "PriRadar: A Privacy-Preserving Framework for Spatial Crowdsourcing," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 299–314, 2019.

[3] J. Zhang, F. Yang, Z. Ma, Z. Wang, X. Liu, and J. Ma, "A Decentralized Location Privacy-Preserving Spatial Crowdsourcing for Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2299–2313, 2020.

[4] B. Gedik and L. Liu, "Location Privacy in Mobile Systems: A Personalized Anonymization Model," in *25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2005, pp. 620–629.

[5] G. Ghinita, K. Zhao, D. Papadias, and P. Kalnis, "A reciprocal framework for spatial K-anonymity," *Information Systems*, vol. 35, no. 3, pp. 299–314, 2010.

[6] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys)*, pp. 31–42, 2003.

[7] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The New Casper: Query Processing for Location Services without Compromising Privacy," *Proceedings of the 32nd International Conference on Very large data bases (VLDB)*, vol. 6, pp. 763–774, 2006.

[8] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 7, no. 10, pp. 919–930, 2014.

[9] Y. Shen, L. Huang, L. Li, X. Lu, S. Wang, and W. Yang, "Towards Preserving Worker Location Privacy in Spatial Crowdsourcing," in *IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.

[10] A. Liu, W. Wang, S. Shang, Q. Li, and X. Zhang, "Efficient task assignment in spatial crowdsourcing with worker and task privacy protection," *GeoInformatica*, vol. 22, no. 2, pp. 335–362, 2018.

[11] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A Survey on Homomorphic Encryption Schemes: Theory and Implementation," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2019.

[12] Y. Lindell, "Secure Multiparty Computation (MPC)," *Communications of the ACM*, vol. 64, no. 1, pp. 86–96, 2020.

[13] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, and Programming (ICALP)*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 4052, pp. 1–12.

[14] P. Samarati and L. Sweeney, "Generalizing Data to Provide Anonymity when Disclosing Information," in *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 1998, pp. 10–1145.

[15] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[16] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional K-Anonymity," in *IEEE 22nd International Conference on Data Engineering (ICDE)*. Atlanta, GA, USA: IEEE, 2006, pp. 25–25.

[17] T. Iwuchukwu, D. J. DeWitt, A. Doan, and J. F. Naughton, "K-Anonymization as Spatial Indexing: Toward Scalable and Incremental Anonymization," in *IEEE 23rd International Conference on Data Engineering (ICDE)*. Istanbul, Turkey: IEEE, 2007, pp. 1414–1416.

[18] Q. Tang, Y. Wu, S. Liao, and X. Wang, "Improving Strict Partition for Privacy Preserving Data Publishing," in *First International Conference on Networking and Distributed Computing*, 2010, pp. 207–212.

[19] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*, 2006, pp. 171–178.

[20] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "PRIVE: Anonymous location-based queries in distributed mobile systems," in *Proceedings of the 16th International Conference on World Wide Web*. Banff Alberta Canada: ACM, 2007, pp. 371–380.

[21] ——, "MobiHide: A Mobile Peer-to-Peer System for Anonymous Location-Based Queries," in *Advances in Spatial and Temporal Databases*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4605, pp. 221–238.

[22] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing Location-Based Identity Inference in Anonymous Spatial Queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, 2007.

[23] Y. Xiao, L. Xiong, and C. Yuan, "Differentially Private Data Release through Multidimensional Partitioning," in *Secure Data Management*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 6358, pp. 150–168.

[24] Q. Miao, Jing, Weipeng, and Song, Houbing, "Differential privacy–based location privacy enhancing in edge computing," *Concurrency and Computation: Practice and Experience*, vol. 31, 2018.

[25] G. Liu, Z. Tang, B. Wan, Y. Li, and Y. Liu, "Differential privacy location data release based on quadtree in mobile edge computing," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 6, 2022.

[26] G. Cormode, M. Procopiuc, E. Shen, D. Srivastava, and T. Yu, "Differentially Private Spatial Decompositions," in *IEEE 28th International Conference on Data Engineering (ICDE)*. IEEE, 2012.

[27] W. Qardaji, W. Yang, and N. Li, "Differentially Private Grids for Geospatial Data," in *IEEE 29th International Conference on Data Engineering (ICDE)*. IEEE, 2013, pp. 757–768.

[28] N. Niknami, M. Abadi, and F. Deldar, "SpatialPDP: A personalized differentially private mechanism for range counting queries over spatial databases," in *4th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2014, pp. 709–715.

[29] S. Shaham, G. Ghinita, R. Ahuja, J. Krumm, and C. Shahabi, "HTF: Homogeneous Tree Framework for Differentially Private Release of Large Geospatial Datasets with Self-tuning Structure Height," *ACM Transactions on Spatial Algorithms and Systems*, vol. 9, no. 4, pp. 25:1–25:30, 2023.

[30] S. N. von Voigt, L. Mehner, and F. Tschorsch, "From Theory to Comprehension: A Comparative Study of Differential Privacy and k-Anonymity," in *Proceedings of the Fourteenth ACM Conference on Data and Application Security and Privacy*, 2024.

[31] R. Cummings and J. Sarathy, "Centering Policy and Practice: Research Gaps Around Usable Differential Privacy," in *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. Atlanta, GA, USA: IEEE, 2023, pp. 122–135.

[32] C. Dwork, N. Kohli, and D. Mulligan, "Differential Privacy in Practice: Expose your Epsilons!" *Journal of Privacy and Confidentiality*, vol. 9, no. 2, 2019.

[33] H. Samet, "The Quadtree and Related Hierarchical Data Structures," *ACM Computing Surveys*, vol. 16, no. 2, pp. 187–260, 1984.

[34] A.-e.-e. A. Hussien, N. Hamza, and H. A. Hefny, "Attacks on Anonymization-Based Privacy-Preserving: A Survey for Data Mining and Data Publishing," *Journal of Information Security*, vol. 04, no. 02, pp. 101–112, 2013.

[35] J. R. Douceur, "The Sybil Attack," in *Peer-to-Peer Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, vol. 2429, pp. 251–260.

[36] L. A. Martucci, M. Kohlweiss, C. Andersson, and A. Panchenko, "Self-certified Sybil-free pseudonyms," in *Proceedings of the First ACM Conference on Wireless Network Security*. Alexandria VA USA: ACM, 2008, pp. 154–159.

[37] E. Makri, D. Rotaru, F. Vercauteren, and S. Wagh, "Rabbit: Efficient Comparison for Secure Multi-Party Computation," in *International Conference on Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 249–270.

[38] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) Fully Homomorphic Encryption without Bootstrapping," *ACM Transactions on Computation Theory*, vol. 6, no. 3, pp. 1–36, 2014.

[39] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs, "Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE," in *Advances in Cryptology - EUROCRYPT 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7237, pp. 483–501.

[40] A. Beimel, "Secret-Sharing Schemes: A Survey," in *Coding and Cryptology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6639, pp. 11–46.

[41] European Commission – Eurostat/GISCO, "Eurostat Census Grid 2021," 2021. [Online]. Available: https://ec.europa.eu/eurostat/web/gisco/geodata/population-distribution/geostat

[42] ——, "The GISCO statistical unit dataset," 2021. [Online]. Available: https://ec.europa.euhttps://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units/territorial-units-statistics