BORDER-BASED ANONYMIZATION METHOD FOR SHARING PRIVATE SPATIAL-TEMPORAL DATA

HANI ABUSHARKH

A THESIS

IN

THE CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE IN INFORMATION SYSTEMS SECURITY

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

SEPTEMBER 2011

© HANI ABUSHARKH, 2011

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Hani AbuSharkh**

Entitled: **Border-based Anonymization Method for Sharing Private Spatial-Temporal Data**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science in Information Systems Security

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. Z. Tian

______________________________ External Examiner
Dr. S. Jahinuzzaman

______________________________ Examiner
Dr. S. Li

______________________________ Supervisor
Dr. B. Fung

Approved ______________________________
Dr. Assi, Chadi , GPD

Concordia Institute for Information Systems Engineering

August 27__________ 2011______ ______________________________
Dr. R. Drew, Dean
Faculty of Engineering and Computer Science

Abstract

Border-based Anonymization Method for Sharing Private Spatial-Temporal Data

Hani AbuSharkh

Many location-based software applications have been developed for mobile devices. Consequently, location-based service providers often have a detailed trajectory history of their service recipients. The collected spatial-temporal information of their service recipients can be invaluable for other organizations and companies in many ways; for example, it can be used for direct marking, market analysis, and consumer behavior analysis. Yet, releasing the spatial-temporal data together with other user-specific data in its raw format often leads to privacy threats to the service recipients. In this thesis, we study the problem of spatial-temporal data publishing with the consideration of preserving both privacy protection and information utility for data mining. The contributions are in twofold. First, we propose a service-oriented architecture to determine an appropriate location-based service provider for a given data request. Second, we present a border-based data anonymization method to transform a raw spatial-temporal data table into an anonymous version that preserves both privacy and information utility. Empirical results suggest that our proposed method can efficiently and effectively preserve the information required for data mining.

Acknowledgments

This dissertation would not have been possible without the guidance and the help of my supervisor, Dr. Benjamin Fung, who in one way or another contributed and extended his valuable assistance at every single level in the completion of this study. Dr. Fung has supported me thoughout my thesis with his patience and knowledge, while allowing me the room to work in my own way. I attribute the level of my Master's degree to his encouragement and effort. One simply could not wish for a better or friendlier supervisor.

I am also very thankful to Maya Sevaldson, Ali Salem, and Khalil Al-Hussaeni for their continuous friendly encouragement and technical support.

Finally, I would like to express my deep gratitude to the Hani Qaddoumi Scholarship Foundation, represesnted by Rana Diab and Afifah Kittaneh, for the financial assistance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Turn on any smartphone and it is not too difficult to identify some location-based applications, such as a navigation system and social-networks applications. These location-based applications not only provide convenient, customized location-based services to the recipients but they also collect a large volume invaluable user-specific spatial-temporal information for the location-based service providers and their partners. Yet, simply sharing the raw data with their partners will compromise the privacy of the service recipients. The objective of this study is to propose a service-oriented architecture together with a privacy-preserving spatial-temporal data anonymization algorithm to preserve both the privacy of the service recipients and the information utility for data mining.

Figure 1 provides an overview of the research problem. The location-based service providers collect and store a large volume of user-specific spatial-temporal data in their private databases. Some third parties (data miners) would like to obtain an appropriate set of spatial-temporal data to perform data mining tasks such as traffic flow analysis and consumer behavior analysis. The first

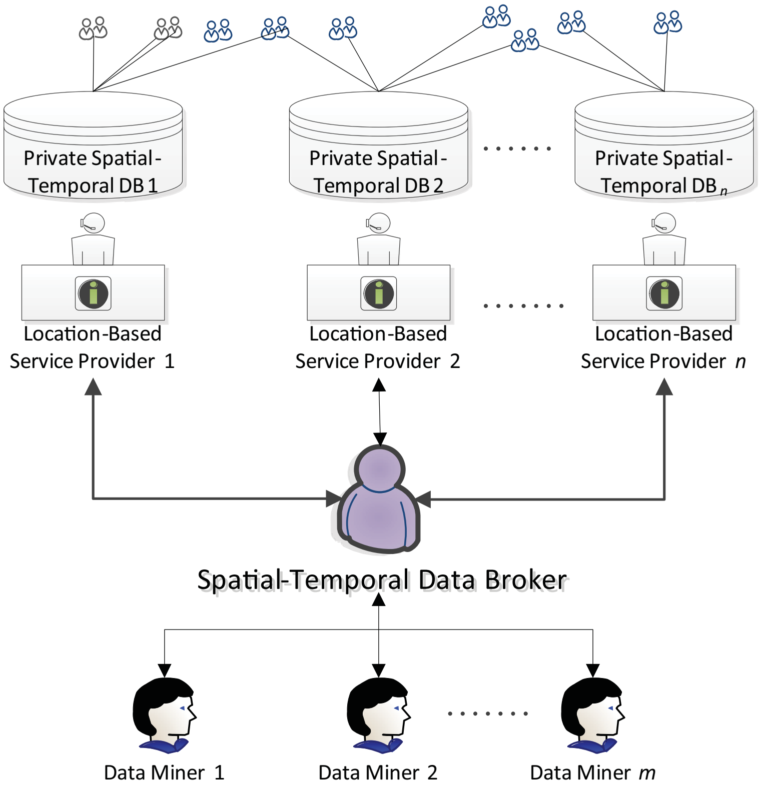

Private Spatial-Temporal DB 1
Private Spatial-Temporal DB 2
Private Spatial-Temporal DB n
Location-Based Service Provider 1
Location-Based Service Provider 2
Location-Based Service Provider n
Spatial-Temporal Data Broker
Data Miner 1
Data Miner 2
Data Miner m

Figure 1: Overview of Spatial-Temporal Data Sharing

challenge is to develop a service-oriented architecture so data miners can identify the location-based service provider(s) who own the data through a service broker. The second challenge is to efficiently anonymize the spatial-temporal data in order to fulfill the data requests. In our model, we do not require the service broker to be a trustworthy entity. Therefore, the data must be anonymized when released by the location-based service providers.

A user-specific spatial-temporal data table contains a collection of spatial-temporal records. Each record consists of a sensitive attribute of a user and a spatial-temporal path that represents the sequence of visited locations with timestamps.

A spatial-temporal data table T is a collection of records in the form $\langle(loc_1t_1) \rightarrow \ldots \rightarrow (loc_nt_n)\rangle : s_1, \ldots, s_p : d_1, \ldots, d_m$, where $\langle(loc_1t_1) \rightarrow \ldots \rightarrow (loc_nt_n)\rangle$ is the spatial-temporal path, $s_i \in S_i$ are the sensitive attributes, and $d_i \in D_i$ are the quasi-identifying attributes (QID) of an object. The sensitive and the QID attributes are object-specific data in the form of relational data. Publishing a table T, which might be for reference or analytical purposes, raises the problem of privacy.

Motivating Example

We use an example to demonstrate the potential privacy risks of releasing the spatial-temporal data in its raw format.

Suppose a location-based service provider wants to share Table 1 with a data miner. Record 2, for example, shows that a user with sensitive value $s3$ visited locations f, h, and e at times 6, 7, and 8, respectively. Without loss of generality, we assume that each record contains only one sensitive attribute in this example.

We identify and address two types of privacy linkage attacks [20]:

Record linkage: a privacy attack in which the adversary exploits the uniqueness of a path in the released data table. If a path is so specific that it only matches a small number of other paths, linking the victim's record from the released spatial-temporal data table and the sensitive value may become possible. The assumption is that the adversary possesses some knowledge about the locations and timestamps (doublets) existing in a victim's path. Suppose one of the data miners is an adversary who knows that the data record of a target victim, Daphne, is in Table 1. This adversary also knows that Daphne has visited $g2$ and $b3$. Daphne's record, together with her sensitive value $s1$, can be uniquely identified because Record 1 is the *only* record that contains $g2$ and $b3$. The adversary can also determine Daphne's other visited locations, such as $d4$, $f6$, and $h7$.

Table 1: Raw Spatial-Temporal Data Table

Rec#	**Path**	**Sensitive**	**...**
1	$\langle g2 \rightarrow b3 \rightarrow d4 \rightarrow f6 \rightarrow h7 \rangle$	s1	...
2	$\langle f6 \rightarrow h7 \rightarrow e8 \rangle$	s3	...
3	$\langle b3 \rightarrow d4 \rightarrow f6 \rightarrow e8 \rangle$	s4	...
4	$\langle g2 \rightarrow c5 \rightarrow h7 \rightarrow e8 \rangle$	s3	...
5	$\langle b3 \rightarrow h7 \rightarrow e8 \rangle$	s4	...
6	$\langle c5 \rightarrow f6 \rightarrow e8 \rangle$	s2	...
7	$\langle g2 \rightarrow f6 \rightarrow h7 \rightarrow e8 \rangle$	s2	...
8	$\langle g2 \rightarrow c5 \rightarrow f6 \rightarrow h7 \rangle$	s1	...

Attribute linkage: a privacy attack that occurs when a group of records that share some combination of doublets contains a frequently appearing sensitive value. Even though a target victim's record might not be identified, inferring the victim's sensitive value from such a group becomes possible. Suppose the adversary knows that Keith has visited $g2$ and $f6$. Since two of the three records (Records 1, 7, 8) containing $g2$ and $f6$ have sensitive value $s1$, the adversary can infer that Keith has $s1$ with $2/3 = 67\%$ confidence.

We do not require the service broker to be a trustworthy entity in our model, which means that the data publisher is not trusted and may attempt to identify sensitive information from record owners. We also assume that the data miner could be an attacker. Thus, the data must be anonymized when released by the location-based service providers.

Spatial-temporal data are very different from traditional relational data due to their special properties:

- High dimensionality: This is an intrinsic characteristic of spatial-temporal data due to the huge number of possible combinations of locations and timestamps. Consider a subway system having 50 stations that operate 20 hours a day. The total number of dimensions of the data table could be 50 x 20 = 1000 dimensions. Each dimension (doublet) could be a potential piece of knowledge used by an adversary to perform record or attribute linkages; therefore, every dimension is considered a potential quasi-identifying (QID) attribute. If we apply a traditional privacy model, such as K*-anonymity*, all dimensions would be included in a single QID and every path would have to be indistinguishable from at least K-1 other paths. In order to achieve K*-anonymity*, the highly dimensional nature of spatial-temporal data would likely cause most of the data to be suppressed. Consequently, the utility of the resultant anonymous data would be insufficient for further data analysis.

- Sparseness: Each path in spatial-temporal data is relatively short. Anonymizing these short, little-overlapping paths in a high-dimensional space poses a significant challenge for traditional anonymization techniques because it is difficult to identify and group the paths together. Enforcing traditional K*-anonymity* on high-dimensional and sparse data would render the data useless.

- Sequentiality: The order of items in each sequence should be kept and considered; for example, $a1 \rightarrow b2$ is different from $b2 \rightarrow a1$. As a result, the number of possible combinations in sequential data is much higher than the number of possible combinations in set-valued data. In addition, in any path, timestamps are always increasing; therefore, sequences such as $a2 \rightarrow a1$ are not valid, as an object cannot go from time 2 to time 1.

In the research area of privacy-preserving data publishing [20], many anonymization algorithms have been proposed to thwart record linkages and attribute linkages in relational data. Yet, the privacy models they employed, such as K-anonymity [38] [41] and its extensions [22] [32] [33] [50] [51], are not applicable to high-dimensional spatial-temporal data [4].

We adopt a novel privacy model, LKC*-privacy*, that addresses the challenges of spatial-temporal data. LKC*-privacy* provides a practical solution to compensate for an adversary's background knowledge.

1.1 Contribution

Our contributions can be summarized as follows: First, based on the practical assumption that an adversary has limited knowledge, we adopt the *LKC-privacy model* [36] to address the special challenges of anonymizing high-dimensional, sparse, and sequential spatial-temporal data. Second, we propose a service-oriented architecture to determine an appropriate location-based service provider for a given data request. Third, we present an efficient border-based anonymization algorithm to achieve *LKC-privacy* while preserving frequent sequences in the anonymous data. Finally, extensive experimental results suggest that our anonymization method is effective for information preservation and is scalable.

1.2 Outline of Thesis

In Chapter 2, we explain the concept of LBS and demonstrate the potential threats to privacy that stem from publishing raw data. Next we present different models proposed to protect the relational data. Then we discuss different techniques developed to anonymize high-dimensional data, and finally we present anonymization methods to protect location privacy.

Chapter 3 formally defines the problem. We define information utility, followed by a description of the LKC*-privacy* model and problem statement.

Chapter 4 presents a service-oriented architecture for privacy-preserving spatial-temporal data sharing.

Chapter 5 presents our proposed anonymizing algorithm, the concept and the construction of the borders, the counting function, and a detailed explanation of the steps in the border-based algorithm.

Chapter 6 presents an empirical study of the algorithm with a focus on information-utility loss and scalability.

And finally, we point out some possible future research directions, and conclude the work in Chapter 7.

Chapter 2

Literature Review

The use of Location-Based Services (LBS) has increased in the last few years. Publishing the collected data in raw format may violate the privacy of service recipients, and many privacy protection studies have been published. In this chapter we explain the concept of LBS and demonstrate the potential threats to privacy that stem from publishing raw data. Next we present different models proposed to protect the relational data. Then we discuss different techniques used to anonymize high-dimensional data, and finally we present anonymization methods to protect location privacy.

2.1 Location-Based Services

Location-Based Service is defined as an information service accessible through the mobile network by mobile devices, and it utilizes the location of the mobile device [44]. OpenGeospatial Consortium defines LBS as a wireless-IP service that uses geographic information to serve a mobile user.

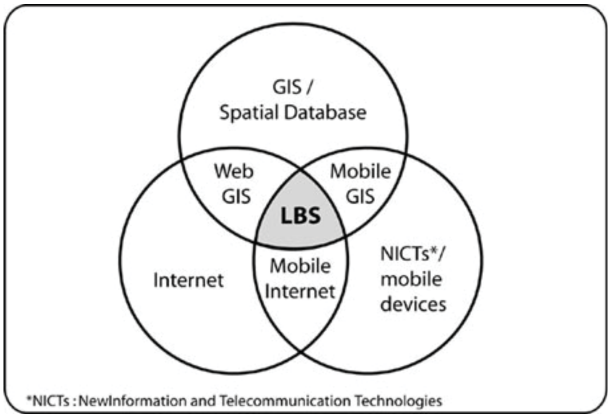


Figure 2: LBS as an Intersection of Technologies [10]

Location and time of an object are usually sent through the Internet to some spatial (or spatial-temporal) database. LBS can be considered as an intersection of three following technologies: a spatial-temporal database, the Internet, and New Information and Communication Technologies (NICTs) [39]. Figure 2 demonstrates the LBS dimensions.

A typical LBS system consists of the following five components [40], as shown in Figure 3:

- *Mobile device:* a pocket-size computing device that can request services from providers. It can be a mobile phone, PDA, or other mobile device. This device sends and receives location and time information about the object.

- *Communication network:* the means through which a mobile device transfers the data back and forth. It can be a Wireless Local Network (WLAN), the Internet, or another network.

- *Positioning components:* a service used to determine the position of user service recipients. Examples include GPS, WLAN, and other communication networks.

- *Service and application provider:* a company or agency that offers a specific service on the network. For example, some mobile applications provide the recipient with the location of

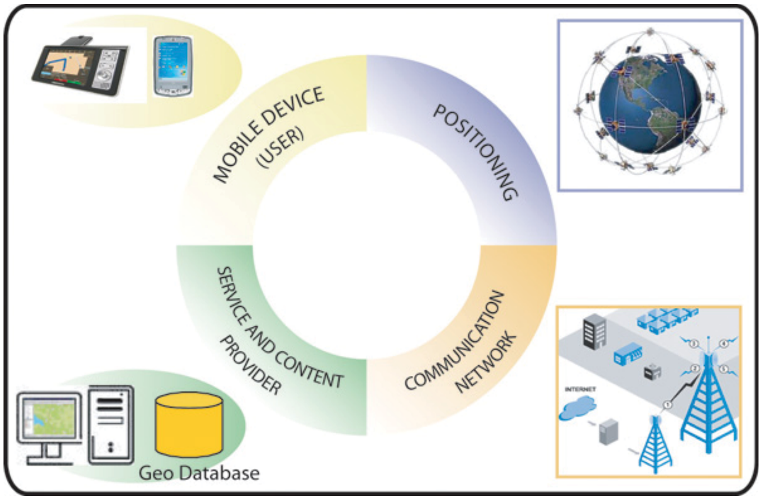


Figure 3: The Basic Components of an LBS [40]

the nearest cinema.

- *Data and Content Provider:* a party who collects the data or information, who may not be the service provider. Service providers may acquire the data from other data and content providers.

The use of LBS has significantly increased in the last few years. LBS applications include tracking, navigation, emergency services, billing, LBS-alerts, social networking, network operator applications, and many end-user applications [47]. However, collecting information about recipients raises many concerns if people are tracked by their positions or by analyzing their preferences and action history. On one hand the analysis helps business applications get a perfect customer model, but on the other raises users' fears about privacy invasion.

In our model, the data collected through different LBSs is passed to a data broker, an independent third party who researches information and data for clients, including different organizations. To address the problem of privacy, the collected data is anonymized before the service provider transmits it to a data broker.

Next we present an overview of the data anonymization approach and the different privacy models that were used to achieve the same purpose.

2.2 Privacy Protection on Relational Data

Many privacy-preserving data publishing techniques have been proposed for anonymizing relational data. We provide a high-level summary of the literature in this section. Relational data are typically stored in tables of the form:

D(Explicit Identifier, Quasi Identifier, Sensitive Attributes,Non-Sensitive Attributes).

Sweeney showed in [41] that even with the removal of *explicit* identifiers, privacy can still be violated through the linking attacks explained in Chapter 1. She indicated a real-life threat to a former governor of the state of Massachusetts. She linked the governor's name in a public voter list with his record in a published medical database by combining the zip code, date of birth, and sex, as shown in Figure 4.

A combination of personal attributes, called a *quasi-identifier*, can be used to identify an individual's record. In the above example, the victim was re-identified by linking his *quasi-identifier* and the victim's record in released data. It was not difficult for the attacker to obtain her boss's zip code, date of birth, and sex, gender which served as the *quasi-identifier*. She also noticed that her boss was hospitalized, and therefore knew that her victim's medical record would appear in the released patient database.

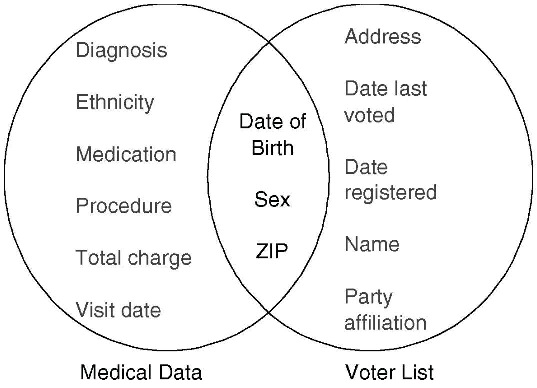


Figure 4: Linking to Re-identify Record Owner [41]

2.2.1 Preserving Threats

Privacy threat occurs when an attacker is able to link a record owner to a record in a published data table, or to a sensitive attribute in a published data table. We call these events record linkage and attribute linkage. In the two types of linkages, we assume that the attacker knows the QID of the victim, we further assume that the attacker knows that the victim's record is in the released table, and seeks to identify the victim's record or sensitive information from the table. A data table is considered to preserve privacy if it can effectively prevent the attacker from successfully performing these linkages.

2.2.2 Privacy Models

Privacy-preserving techniques on spatial-temporal data can be broadly grouped into two categories: data collection and data sharing. While the work on data collection focuses on the privacy and security issues of the sensors and readers at the communication level [30], the work in the data sharing phase focuses on privacy and utility at the data level [25].

In traditional K-anonymity [38] [41], if one record in the table has some value qid, at least

k-1 other records also have the value qid. In other words, the minimum group size on QID is at least k. A table satisfying this requirement is called k-anonymous. In a k-anonymous table, each record is indistinguishable from at least k- 1 other records with respect to QID. Consequently, the probability of linking a victim to a specific record through QID is at most 1/k.

Machanavajjhala et al [33] propose a diversity model called ℓ-diversity. The purpose is to prevent attribute linkage. The ℓ-diversity requires every qid group to contain at least ℓ *well-represented* sensitive values. However, this model cannot prevent probabilistic inference attacks because some sensitive values are naturally more frequent than others in a group, enabling an attacker to conclude that a record in the group is very likely to have those values.

Confidence bounding [46] attempts to prevent attribute linkage. It bounds the confidence of inferring a sensitive value from a qid group by specifying one or more *privacy templates* of the form $\langle QID \rightarrow s, h\rangle$; s is a sensitive value, QID is a quasi-identifier, and h is a threshold. Let $Conf(QID \rightarrow s)$ be $max\{conf(qid \rightarrow s)\}$ over all qid groups on QID, where $Conf(QID \rightarrow s)$ denotes the percentage of records containing s in the qid group. A table satisfies $\langle QID \rightarrow s, h\rangle$ if $Conf(QID \rightarrow s) \leq h$. In other words, $\langle QID \rightarrow s, h\rangle$ bounds the attacker's confidence of inferring the sensitive value s in any group on QID to the maximum h.

The (α, k)-anonymity [50] privacy model requires every qid in a Table T to be shared by at least k records and $Conf(qid \rightarrow s) \leq \alpha$ for any sensitive value s, where k and α are data publisher-specified thresholds. However, it does not limit an adversary's knowledge, which results in high utility loss of data; additionally, it may result in high distortion if the sensitive values are skewed.

[51] proposes the notion of personalized privacy to allow each record owner to specify her own privacy level. This model assumes that each sensitive attribute has a taxonomy tree and that each

record owner specifies a guarding node in this tree. However, record owner privacy is violated if an attacker is able to infer any domain-sensitive value within the subtree of her guarding node with a probability, called *breach probability*, greater than a certain threshold.

Various cryptographic solutions [55], anonymous communications [11] [29], and statistical methods [48] have been proposed in which only authorized and trustworthy recipients are given the private key to access the data. However, it is difficult to guarantee that all staff in a given company are trustworthy. Our assumption defines the problem and solutions differently from the encryption and cryptographic approaches: Data miners and recipients are not trustworthy; thus, the data must be anonymized when released by the location-based service providers.

2.2.3 Anonymization Operations

Typically, a raw data table does not satisfy a specified privacy requirement and the table must be modified before being published.

The modification is composed of a sequence of anonymization operations that can be broadly divided into three categories as follows:

- Generalization and Suppression: Generalization and suppression aims at hiding some details in QID. Generalization replaces some values with a parent value in the taxonomy of an attribute.

  In a full-domain generalization scheme [31] [38] [41], all values in an attribute are generalized to the same level of the taxonomy tree. In a Subtree generalization scheme [7] [22] [28] [31] [21], at a non-leaf node either all child values or none are generalized. The sibling generalization scheme [31] is similar to the subtree generalization except that some siblings may remain ungeneralized. A parent value is then interpreted as representing all missing child values. In the cell generalization scheme [31] [49] [52], also known as local recoding, some instances of a value may remain ungeneralized while other instances are generalized.

Suppression on the other hand removes or deletes any value that can be used to launch the attack. There are also several different suppression schemes: Record suppression [7] [28] [31] [38] refers to suppressing an entire record. Value suppression [45] [46] refers to suppressing every instance of a given value in a table. Cell suppression (or local suppression) [13] [34] refers to suppressing some instances of a given value in a table.

- Anatomization and Permutation: Anatomization [51], unlike generalization and suppression, does not modify the quasi-identifier or the sensitive attribute, but instead deassociates the relationship between the two. Precisely, the method releases the data on QID and the data on the sensitive attribute in two separate tables: a quasi-identifier table QIT contains the QID attributes, a sensitive table ST contains the sensitive attributes, and both QIT and ST have one common attribute, $GroupID$. All records in the same group will have the same value of $GroupID$ in both tables, and are therefore linked to the sensitive values in the group in exactly the same way. If a group has ℓ distinct sensitive values and each distinct value occurs exactly once in the group, then the probability of linking a record to a sensitive value by GroupID is $1/\ell$. The attribute linkage attack can be distorted by increasing ℓ.

  Permutation shares the same spirit of anatomization. In [57], Zhang et al. proposed an approach called permutation. The idea is to deassociate the relationship between a quasi-identifier and a numerical sensitive attribute by partitioning a set of data records into groups and shuffling their sensitive values within each group.

- Perturbation: The general idea is to replace the original data values with synthetic data values so that the statistical information computed from the perturbed data does not significantly differ from the statistical information computed from the original data. The perturbed data records do not correspond to real-world record owners, so the attacker cannot perform sensitive linkages or recover sensitive information from the published data.

2.3 Anonymizing High-Dimensional Data

As discussed in Chapter 1, temporal-spatial data is high-dimensional. K-anonymity and the other privacy-preserving models mentioned so far suffer from the curse of high dimensionality [4] and render the high-dimensional data useless for data mining.

In this thesis, we solve the problem of high dimensionality by assuming that the adversary knows at most L doublets of a victim's locations and the corresponding times. Mohammed [36] proposed a *LKC-privacy* model that addresses the privacy issues on high-dimensional *relational data*. In contrast, this thesis proposes an anonymization algorithm to achieve the LKC-privacy model on spatial-temporal data. Furthermore, none of the tested traditional QID-based anonymization methods mentioned above, namely [22] [31] [33], are scalable to handle the high-dimensional data in our experiments. K-anonymity [38] [41], confidence bounding [46], and (α, k)-anonymity [50] are special cases of the LKC-privacy model; our anonymization algorithm can also be viewed as a scalable solution for achieving these traditional privacy models. Dwork [16] proposed a privacy model called *differential privacy* that ensures that the removal or addition of a single data record does not significantly affect the overall privacy of the database. Most of the works in differential privacy are based on the interactive privacy model, where the result of a query is in the form of aggregation [14] [17]. Yet, differential privacy may not prevent linkage attacks.

Other recent work has focused on anonymizing high-dimensional transaction data [3] [24] [43] [53] [54]. In [3] Aggarwal and Yu formalized an anonymity model for the sketch-based approach, and utilized it to construct sketch-based privacy-preserving representations of the original data. The sketch-based approach [6] reduces the dimensionality of the data by generating a new representation with a much smaller number of features, where each one uses a different set of random weights to produce a weighted sum of the original feature values. This technique is quite effective for high-dimensional data sets, as long as the data is sparse. The sketch-based method provides privacy protection while allowing effective reconstruction of many aggregate distance measures. Therefore, it can be used for a variety of data mining algorithms such as clustering and classification.

The models suggested in [53] and [54] limit the adversary's power by a maximum number of known $items$ as background knowledge in order to solve the problem of high dimensionality. This assumption is similar to ours, but our problem has two major differences. First, a transaction is a *set* of items, but a spatial-temporal path is a *sequence* of visited location-time doublets. Sequential data drastically increases the computational complexity for counting the support counts as compared to transaction data. Hence, their proposed models are not applicable to spatial-temporal data. Second, we have different privacy and utility measures. The privacy model of [43] is based on only K-*anonymity* and does not consider attribute linkages. [53] [54] measure their data utility in terms of preserved *item instances* and *frequent itemsets*, while we measure the utility based on the number of preserved *frequent sequences*. Xu et al. [53] use a border-based method, but it was not used for sequential or spatial-temporal data.

2.4 Location Privacy

Location anonymity is achieved by mixing the user's identity and request with those of other users. Examples of such techniques are Mix Zones [9], cloaking [26], and location-based k-anonymity [23]. The objectives of these techniques are very different from the solution presented in this thesis. First, their goal was to anonymize an individual user's identity resulting from a set of LBS requests, but our goal is to anonymize high-dimensional data. Second, they dealt with small dynamic groups of users, but we anonymize a large static data set. Hence, their problem is very different from that of spatial-temporal data publishing.

In [2], Papadimitriou et al studied the privacy issue in publishing time-series data and examined the trade-offs between time-series compressibility and partial information hiding and their fundamental implications for how one should introduce uncertainty about individual values by perturbing them. The study found that by making the perturbation *similar* to the original data, we can both preserve the structure of the data better, and simultaneously make breaches harder. However, as data becomes more compressible, a fraction of the uncertainty can be removed if true values are leaked, revealing how they were perturbed.

2.5 Anonymizing Moving Objects

Moving-object data poses new challenges to traditional database, data mining, and privacy-preserving technologies due to its unique characteristics: time-dependency, location-dependency, and high dimensionality. Some recent works [19] [27] [37] [42] [56] address the anonymity of moving objects.

Abul et al. [1] proposed a new privacy model called (k, δ)-*anonymity* that is an extension of traditional K-anonymity. It exploits the inherent uncertainty of moving objects' locations. Their method relies on a basic assumption that every path is continuous. In the 3-dimensional representation, a path is a polyline (x, y, t) where the coordinates (x, y, t) of each point in the polyline represent the moving object's location(x, y) at a specific time t. A minimum of k objects should appear within the radius of δ of the path of every moving object in the same period of time. To achieve the previous privacy requirement for a target data set, [2] uses space translation to change the location coordinates of some points on certain polylines. Figure 5 depicts the model. Although this assumption is valid for some spatial-temporal devices where the object can be traced all the time, it does not hold for others. Another major difference is that this model achieves anonymity by space translation that changes the actual location of an object. In contrast, our approach employs suppression for anonymity and thus preserves the data truthfulness and frequent sequences with true support counts.

In [27] an uncertainly-aware privacy algorithm for GPS traces is presented. The researchers selectively removed doublets to increase uncertainly between paths to hinder identification. The works target GPS-only traces and cannot be employed for anonymizing other spatial-temporal data, so the mechanism cannot be generalized for all spatial-temporal data.

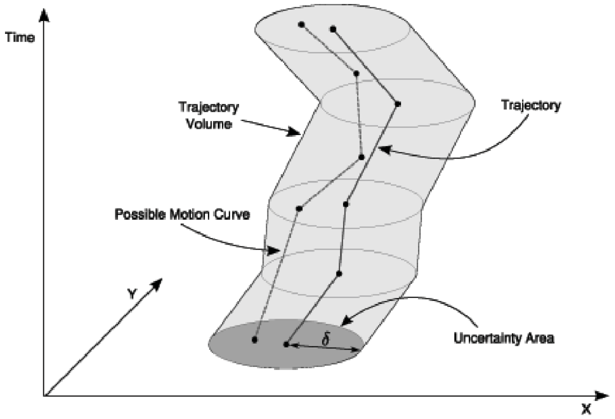


Figure 5: A Graphical Representation of an Uncertainty Trajectory Volume [1]

The privacy model proposed in [42] assumes that different adversaries have different background knowledge about the spatial-temporal paths, and thus their objective is to prevent adversaries from gaining any further information from the published data. They consider the locations in a path as sensitive information and assume that the data holder has the background knowledge of all the adversaries. In reality, such complete information is difficult to obtain.

Pensa et al. [37] proposed a k-anonymity notion for sequence datasets. The proposed algorithm also aims to preserve frequent sequential patterns. However to achieve anonymity, they transformed a sequence into the other by insertion, deletion, or substitution of a single item. Thus, their approach also spoils data truthfulness.

Yarovoy et al. [56] consider time as a QID attribute. However, there is no fixed set of times for all moving objects. Each spatial-temporal path has its own set of times as its QID. It is unclear how the data holder could determine the QID attributes for each trajectory.

Fung et al. [18] [19] and Chen et al. [12] propose a method for anonymizing spatial-temporal data without preserving frequent sequences. Mohammed et al. [35] present a tree-based anonymization method to preserve both privacy and maximal frequent sequences. In contrast, this thesis presents a border-based method and aims at preserving all frequent sequences.

Chapter 3

Problem Definition

A data miner wants to perform a data mining task on a user-specific spatial-temporal dataset, and wants to obtain the dataset from some location-based data service providers. The data miner specifies a data request to a spatial-temporal data broker who is responsible for identifying the location-based service providers who own the requested data. The research problem is how to transform the raw spatial-temporal data (e.g., Table 1) into an anonymous version that simultaneously preserves both the privacy of the underlying users and utility of the information for data mining.

In this chapter, we first define the format of user-specific spatial-temporal data, followed by the LKC-privacy model [36], and the information utility measure in the context of spatial-temporal data. Finally, we provide the problem statement.

3.1 User-Specific Spatial-Temporal Data

We formally define the format of a user-specific spatial-temporal data table as follows: A user-specific spatial-temporal dataset is a collection of spatial-temporal records in which each user's record consists of four types of information:

- *Explicit identifiers* are attributes that can uniquely identify a user, e.g., name, SSN, and phone number.
- *Quasi-identifiers* (*QID*) are some combinations of QID values may identify a user, e.g., job, age, and gender. They are not explicit identifiers.
- *Sensitive attributes* contain some sensitive information that a user may not want other people to know, e.g., disease and financial status.
- *Spatial-temporal information* is a sequence of locations visited by a user within a period of time, e.g., $\langle a1 \rightarrow c4 \rightarrow b7 \rangle$.

Explicit identifiers, e.g., name, SSN, and ID, are removed. Note, we keep the ID in our examples for discussion purposes only. The spatial-temporal data, the user-specific QID, and sensitive attributes are assumed to be important for the data mining task; otherwise, they should be removed.

Definition 3.1.1 (Doublet) A *doublet*, denoted by $\ell_i time_i$, is a combination of location ℓ_i and timestamp t_i.

Definition 3.1.2 (Spatial-temporal path) A *spatial-temporal path*, denoted by $\langle \ell_1 t_1, \ldots, \ell_n t_n \rangle$, is a sequence of doublets sorted by the timestamps in ascending order, representing a sequence of locations visited by a user between timestamps t_1 and t_n, inclusively.

A timestamp is the entry time to a location. A user is assumed to be staying in the same location until detected again in another location. A user may revisit the same location at different times but consecutive doublets having the same location are considered redundant and, therefore, are removed. For example, in $\langle c1 \rightarrow c2 \rightarrow c3 \rightarrow d4 \rightarrow c5\rangle$, $c2$ and $c3$ are removed but $c1$, $d4$, and $c5$ are kept. At any time, a user can appear in at most one location, so $\langle a1 \rightarrow b1\rangle$ is not a valid spatial-temporal path. In other words, the timestamps increase monotonically in a spatial-temporal path.

Definition 3.1.3 (User-specific spatial-temporal table) A *user-specific spatial-temporal table* (or simply a spatial-temporal table) T consists of a collection of records in the form $[\langle \ell_1 t_1, \ldots, \ell_n t_n\rangle,$ $a_1, \ldots, a_y,\ s_1, \ldots, s_m]$, where $\langle \ell_1 t_1, \ldots, \ell_n t_n\rangle$ is a spatial-temporal path, $a_1, \ldots, a_y$ are quasi-identifying values, and $s_1, \ldots, s_m$ are sensitive values of a user.

3.2 Information Utility

The measure of information utility varies depending on the data mining task to be performed on the published data. In this thesis, we aim at preserving the frequent sequences. A sequence $q = \langle \ell_1 t_1 \rightarrow \ldots \rightarrow \ell_n t_n\rangle$ is an ordered set of doublets.

Definition 3.2.1 (Frequent sequence) A sequence q is *frequent* in a spatial-temporal table T if $|T(q)| \geq K'$, where $T(q)$ is the set of records in T containing q and K' is a minimum support threshold.

$F(T)$ denotes the set of frequent sequences in T with respect to K'. Frequent sequences capture the major traffic flows [8] and often form the information basis for different primitive data mining tasks on sequential data, e.g., association-rules mining [5]. In the context of spatial-temporal data, association rules can be used to determine the subsequent locations of a moving object given the previously visited locations. This knowledge is important for workflow mining [25].

If q is a frequent sequence, every subsequence p with $p \preceq q$ is also a frequent sequence.

3.3 Privacy Model

One data miner, who is an adversary, seeks to identify the record or sensitive values of some target victim in T. As explained earlier, we assume that the adversary knows at most L doublets that the victim has previously visited. We use ρ to denote such a prior known sequence of doublets, where $|\rho| \leq L$. Based on the prior knowledge ρ, the adversary could identify a group of records, denoted by $T(\rho)$, that “contains" ρ. A record in T *contains* ρ if ρ is a subsequence of the spatial-temporal path in the record. For example in Table 1, Records #1,2,7,8 contain $\rho = \langle f6 \rightarrow h7 \rangle$, written as $T(\rho) = \{Records\#1, 2, 7, 8\}$. Based on $T(\rho)$, the adversary could launch two types of privacy attacks.

The first type of privacy attack is *record linkage*. Given prior knowledge ρ, $T(\rho)$ is a set of candidate records that contains the victim's record. If the group size of $T(\rho)$, denoted by $|T(\rho)|$, is small, then the adversary may identify the victim's record from $T(\rho)$ and, therefore, the victim's sensitive value. For example, if $\rho = \langle g2 \rightarrow b3 \rangle$ in Table 1, $T(\rho) = \{Record\#1\}$. Thus, the adversary can easily infer that the victim's sensitive value is $s1$.

The second type of privacy attack is *attribute linkage*. Given prior knowledge ρ, the adversary can identify $T(\rho)$ and infer that the victim has sensitive value s with confidence $P(s|\rho) = \frac{|T(\rho \wedge s)|}{|T(\rho)|}$, where $T(\rho \wedge s)$ denotes the set of records containing both ρ and s. $P(s|\rho)$ is the percentage of records in $T(\rho)$ containing s. The privacy of a victim is at risk if $P(s|\rho)$ is high. For example, given $\rho = \langle g2 \rightarrow f6 \rangle$ in Table 1, $T(\rho \wedge s1) = \{Records\#1, 8\}$ and $T(\rho) = \{Records\#1, 7, 8\}$; therefore, $P(s1|\rho) = 2/3 = 67\%$.

To thwart the record and attribute linkages, we adopt the *LKC-privacy model* [36], which was originally proposed for relational data, and we apply the model in the context of spatial-temporal data. Intuitively, LKC-privacy requires that every sequence with a maximum length L in the spatial-temporal table has to be shared by at least a certain number of records, and the ratio of sensitive value(s) in every group cannot be too high.

Definition 3.3.1 (LKC-privacy) Let L be the maximum length of the prior knowledge. Let S be a set of sensitive values. A spatial-temporal data table T satisfies *LKC-privacy* if and only if for any non-empty sequence q with $|q| \leq L$ of any spatial-temporal path in T,

1. $|T(q)| \geq K$, where $K > 0$ is an integer anonymity threshold, and

2. $P(s|q) \leq C$ for any $s \in S$, where $0 < C \leq 1$ is a real number confidence threshold.

A location-based service provider specifies the thresholds L, K, and C. The maximum length L reflects the assumption of the adversary's power. LKC-privacy guarantees the probability of a successful identity linkage to be $\leq 1/K$ and the probability of a successful attribute linkage to be $\leq C$.

Intuitively, a sequence q with $|q| \leq L$ is a violation in T with respect to a given LKC-privacy requirement if $T(q)$ violates at least one of the conditions in Definition 3.3.1.

Definition 3.3.2 (Violating sequence) Let q be a sequence of a spatial-temporal path in T with $0 \leq |q| \leq L$. q is a *violating sequence* with respect to a LKC-privacy requirement if $|T(q)| < K$ or $P(s|q) > C$ for any sensitive value $s \in S$. $V(T)$ denotes the set of violating sequences in T with respect to a LKC-privacy requirement.

Observation 3.3.1 If q is a violating sequence, every supersequence p with $q \preceq p$ and $|q| \leq L$ is also a violating sequence.

Example 3.3.1 Let $L = 2$, $K = 2$, $C = 50\%$, and $S = \{s1\}$. In Table 1, a sequence $q_1 = \langle g2 \rightarrow d4 \rangle$ is a violating sequence because $|T(q_1)| = 1 < K$. A sequence $q_2 = \langle g2 \rightarrow f6 \rangle$ is a violating sequence because $P(s1|q_2) = 67\% > C$. However, a sequence $q_3 = \langle g2 \rightarrow c5 \rightarrow f6 \rightarrow h7 \rangle$ is not a violating sequence even if $|T(q_3)| = 1 < K$ and $P(s1|q_3) = 100\% > C$ because $|q_3| > L$.

In order to achieve *LKC-privacy*, it is not correct to ignore sequences of size less that L. In other words, if a table T satisfies *LKC-privacy* it does not mean it satisfies *LḰC-privacy*, where $L' < L$.

Lemma 1 *LKC-privacy* is not monotonic with respect to adversary's knowledge L.

Proof. To prove that LKC-privacy is not monotonic with respect to L, it is sufficient to prove that one of the conditions of *LKC-privacy* in Definition 1 is not monotonic. In the following we provide a counterexample for both conditions:

Condition 1: K is not monotonic with respect to L. In other words, If all of the size-L sequences are nonviolating, it does not guarantee that a sequence with size $L' \leq L$ is also nonviolating. For example, in Table 2, though the size-3 sequences satisfy privacy requirement for K = 2, the size-2 sequence, $q = \langle a1 \rightarrow d2 \rangle$ does not satisfy the requirement.

Table 2: Counterexample for Monotonic Property

ID	Path	Sensitive	...
1	$\langle a1 \rightarrow d2 \rangle$	s1	...
2	$\langle a1 \rightarrow b2 \rangle$	s3	...
3	$\langle a1 \rightarrow b2 \rightarrow c3 \rangle$	s3	...
4	$\langle a1 \rightarrow b2 \rightarrow c3 \rangle$	s4	...

Condition 2: C is not monotonic with respect to L. If q is a nonviolating sequence with $P(s|q) \leq C$ and $|T(q)| \geq K$, its subsequence q' may or may not be a nonviolating sequence. For example in Table 2, the sequence $q = \langle a1 \rightarrow b2 \rightarrow c3 \rangle$ satisfies $P(s3|q)$ = 50 $\leq C$. However, its subsequence $q' = \langle a1 \rightarrow b2 \rangle$ does not satisfy $P(s3|q') = 100\% > C$.

Therefore, in ***LKC-privacy***, it is not sufficient to assure that every sequence q satisfies both conditions given length L in T. Instead, we have to ensure that every sequence q with length not greater than L satisfies the conditions. To overcome this bottleneck, we suppress the minimal violating sequences that exist within the violating sequences which sufficient to satisfy the ***LKC-privacy*** model.

In our model, we use *global suppression* [20] to remove the violating sequences. Global suppression means deleting the item from all transactions that contain the item. Such item suppression has the following properties:

1. Suppressing an item eliminates all sequences that contain the item.

2. Suppressing an item does not alter any sequence and its support that does not contain the item.

3. Suppressing an item does not introduce a new sequence.

We adopt the LKC-privacy model because it has the following desirable properties that are important for anonymizing high-dimensional data:

The data holder has the capability to determine the values of L, K, and C. This gives the data holder the flexibility to determine the level of privacy based on the holder's needs.

LKC-privacy guarantees the probability of a successful identity linkage to be $\leq 1/K$, and the probability of a successful attribute linkage to be $\leq C$.

The LKC privacy model is flexible to adjust the trade-off between data privacy and data utility, and between an adversary's power and data utility. Increasing L and K, or decreasing C, would improve the privacy at the expense of data utility.

LKC-privacy generalizes several traditional privacy models. K-anonymity [38] [41] is a special case of LKC-privacy with $C = 100\%$ and $L = |d|$, where $|d|$ is the number of dimensions, i.e., number of distinct doublets, in the table. Confidence bounding [46] is a special case of LKC-privacy with $K = 1$ and $L = |d|$. (a, k)-anonymity [49] is also a special case of LKC-privacy with $L = |d|$, $K = k$, and $C = a$. Thus, the data holder can still achieve the traditional models, if needed.

LKC-privacy is a general privacy model that thwarts both identity linkage and attribute linkage. It is also a privacy model that is applicable to anonymize spatial-temporal data with or without sensitive attributes.

3.4 Problem Statement

The research problem studied in this thesis can be summarized in two subproblems:

1. Given a data mining request, the problem is to develop an effective service-oriented architecture to determine the appropriate location-based service providers who own the data that satisfies the data mining request, and to establish a connection session between each service provider and the data miner.

2. Given a spatial-temporal table T, a LKC-privacy requirement, a minimum support threshold K', and a set of sensitive values S, the problem is to identify a transformed version of T that satisfies the LKC-privacy requirement while preserving the maximum number of frequent sequences $|F(T)|$.

The information sharing process can be divided into two phases. In the first phase (Chapter 4), the data broker receives requests from data miners and establishes connections with the location-based service providers who contribute their data in a privacy-preserving manner. In the second phase (Chapter 5), the location-based service providers anonymize their spatial-temporal data based on their own privacy requirement and the data miner's information utility requirement, and then they submit the anonymous data to the data broker, who will then pass it to the data miner.

Chapter 4

Service-Oriented Architecture (SOA) for Sharing Private Spatial-Temporal Data

In this chapter we present a service-oriented architecture for sharing private spatial-temporal data. Figure 6 depicts an overview of the communication channels of the participants. Given a data request, the *data broker* plays the central role in identifying the contributing location-based service providers and presenting the final anonymous data to the data miner. The architecture does *not* require the data broker to be a trusted entity. This makes our architecture practical because a trusted party is often not available in real-life scenarios.

The objective of this phase is to establish a common session context among the contributing location-based service providers and the data miner in four steps: *data miner authentication*, *contributing data providers identification*, *session initialization*, and *requirements negotiation*.

Data miner authentication: The data broker first authenticates a data miner to the requested service, generates a session token for the current interaction, and then identifies the location-based

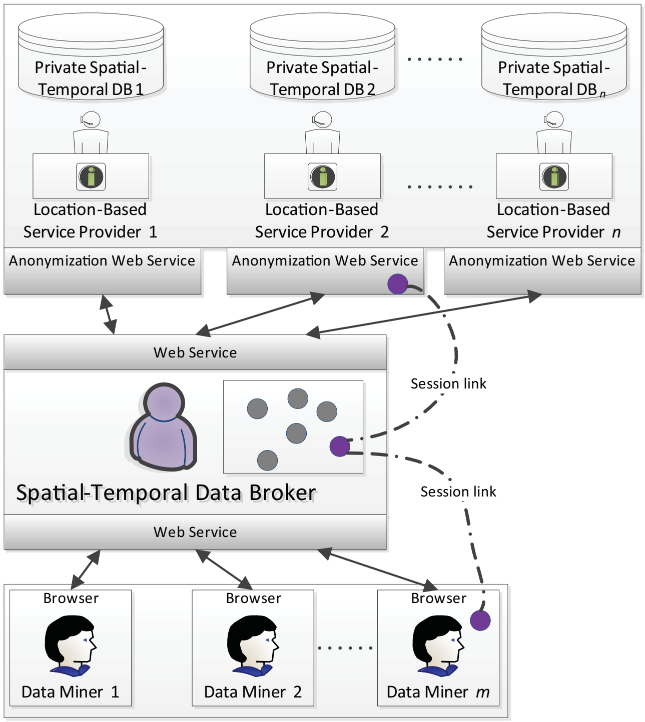

Private Spatial-Temporal DB 1
Private Spatial-Temporal DB 2
Private Spatial-Temporal DB n
Location-Based Service Provider 1
Location-Based Service Provider 2
Location-Based Service Provider n
Anonymization Web Service
Anonymization Web Service
Anonymization Web Service
Web Service
Spatial-Temporal Data Broker
Web Service
Session link
Session link
Browser
Browser
Browser
Data Miner 1
Data Miner 2
Data Miner m

Figure 6: Service-Oriented Architecture for Privacy-Preserving Spatial-Temporal Data Sharing

service providers accessible by the data miner.

Contributing service providers identification: Next, the data broker queries the data schema of the accessible location-based service providers to identify which can contribute data for the requested service. To facilitate more efficient queries, the broker could periodically pre-fetch data schema from the service providers, or the providers could update their data with the broker.

Session initialization: Next, the data broker notifies all contributing location-based service providers with the session identifier. All prospective service providers share a common session context that represents a stateful presentation of information related to a specific execution of a spatial-temporal data anonymization algorithm called *ST-Anonymizer*, which will be discussed in Chapter 5. Due to the fact that multiple parties are involved and the flow of multiple protocol messages is needed in order to fulfill the data integration, we use *Web Service Resource Framework*

(*WSRF*) to keep the stateful information along an initial data request. An established session context stored as a single web service resource contains several attributes to identify a ST-Anonymizer process with *end-point reference* (*EPR*), the client address, the data providers' addresses and their certificates, and an authentication token that contains the data miner's certificate.

Requirements negotiation: The data broker is responsible for communicating the negotiation of privacy and information requirements among the data providers and the data miner. Specifically, this step involves negotiating the price, the privacy requirement in Definition 3.3.1, and the minimum support threshold K'.

Chapter 5

Spatial-Temporal Data Anonymization

A spatial-temporal data table satisfies a given LKC-privacy requirement if all violating sequences with respect to a given LKC-privacy requirement are removed. The objective of this phase is to anonymize a spatial-temporal table by eliminating all violating sequences while preserving as many frequent sequences as possible. We first present the algorithm used to compute the *violating sequences*; followed by the suppression algorithm; then, a border-based representation of the violating sequences, frequent sequences, and the counting function; finally, we brief the steps of a border-based suppression algorithm.

5.1 Computing Violating Sequences

Lemma 2 A spatial-temporal data table T satisfies *LKC-privacy* if and only if T contains no MVS. ■

$Proof$. Suppose a data table T does not satisfy *LKC-privacy* even if it contains no MVS. Then, by Definition, table T contains a violating sequence. But, a violating sequence must be a MVS, or its subset is MVS, which contradicts the initial assumption. Therefore, the data table T must satisfy *LKC-privacy*.

Algorithm 1 presents a method to efficiently generate all the minimal violating sequences based on a *LKC-Privacy* model. Line 1 puts all the size-1 sequences, i.e., all distinct doublets, as candidates X_1 of MVS. Line 4 scans T once to compute $|T(q)|$ and $BPr(s|q)$ for each sequence $q \in X_i$ and for each sensitive value $s \in S$. If the sequence q violates the LKC-privacy requirement in Line 6, then we add q to the minimal violating sequences set V_i (Line 7); otherwise, we add q to the non-violating sequence set W_i (Line 9) for generating the next candidate set X_{i+1}, which is a self-join of W_i (Line 12). Two sequences $q_x = \langle(loc^x_1 t^x_1) \rightarrow \ldots \rightarrow (loc^x_i t^x_i)\rangle$ and $q_y = \langle(loc^y_1 t^y_1) \rightarrow \ldots \rightarrow (loc^y_i t^y_i)\rangle$ in W_i can be joined only if the first $i-1$ doublets of q_x and q_y are identical and $t^x_i < t^y_i$. The joined sequence is $\langle(loc^x_1 t^x_1) \rightarrow \ldots \rightarrow (loc^x_i t^x_i) \rightarrow (loc^y_i t^y_i)\rangle$. Lines 13-17 remove a candidate q from X_{i+1} if q is a supersequence of any sequence in V_i because any proper subsequence of a MVS cannot be a violating sequence. The set of VS, denoted by $V(T)$, is the union of all V_i.

Example 5.1.1 Consider Table 1 with $L = 2$, $K = 2$, and $C = 50\%$. Suppose $X_1 = \{g2, b3, h7, e8\}$. After scanning T, we divide X_1 into $V_1 = \emptyset$ and $W_1 = \{g2, b3, h7, e8\}$. Next, from W_1 we generate the candidate set $X_2 = \{g2b3,\ g2h7,\ g2e8,\ b3h7,\ b3e8,\ h7e8\}$. We scan T again to determine $V_2 = \{g2b3,\ b3e8\}$. We do not further generate X_3 because $L = 2$. ■

Definition 5.1.1 (Violating doublet) A doublet p is a *violating doublet* if it is part of a violating sequence. ■

Algorithm 1 Generate minimal violating sequences

```
**Input:** Raw spatial-temporal data table $T$
**Input:** Thresholds $L$, $K$, and $C$
**Input:** Sensitive values $S$
**Output:** minimal violating sequence $V(T)$
$X_1 \leftarrow$ set of all distinct doublets in $T$;
$i = 1$;
while $i \leq L$ and $X_i \neq \emptyset$ do
    Scan $T$ to compute $|T(q)|$ and $BPr(s|q)$, for $\forall q \in X_i$, $\forall s \in S$;
    for $\forall q \in X_i$ where $|T(q)| > 0$ do
        if $|T(q)| < K$ or $BPr(s|q) > C$ then
            Add $q$ to $V_i$;
        else
            Add $q$ to $W_i$;
        end if
    end for
    $X_{i+1} \leftarrow W_i \bowtie W_i$;
    for $\forall q \in X_{i+1}$ do
        if $q$ is a super sequence of any $v \in V_i$ then
            Remove $q$ from $X_{i+1}$;
        end if
    end for
    $i$++;
end while
return $V(T) = V_1 \cup \ldots \cup V_{i-1}$;
```

Example 5.1.2 Given the set of minimal violating sequence, $V(T) = \{g2b3, b3e8\}$, the violating doublets are $\{g2, b3, e8\}$. ■

We have to remove all the violating sequences to satisfy the LKC-privacy requirement. We can remove all the minimal violating sequences by suppressing a subset of violating doublets. Given, $V(T) = \{g2b3, b3e8\}$, we can either suppress $\{b3\}$, or $\{g2, e8\}$, and so on.

Algorithm 2 ST-Anonymizer

```
$Supp = \emptyset$;
while $|V(T)| > 0$ do
    Select a doublet $d$ with the maximum $Score(d)$;
    $Suppress$ d;
    Update $Score(d')$ if any sequence in $V(T)$ or $F(T)$ containing both $d$ and $d'$;
end while
return Table $T$ after suppressing doublets in $Supp$;
```

5.2 Spatial-Temporal Anonymizer

The elimination of violating sequences is achieved by suppressing a subset of doublets from the table. Specifically, we employ the *global suppression* scheme [20] that was explained in chapter 3. Algorithm 2 provides an overview of the *Spatial-Temporal (ST)-Anonymizer*. The algorithm iteratively selects a doublet d for suppression based on goodness function $Score(d)$, updates the $Score(d)$ of remaining doublets, and terminates when all violating sequences have been eliminated.

Intuitively, we prefer suppressing a doublet d that maximizes the number of eliminated violating sequences and minimizes the number of eliminated frequent sequences for suppression. Thus, we define a greedy function $Score(d)$ that quantifies the goodness of suppression of a doublet d with respect to the number of eliminated violating sequences $|V(d)|$ and the number of eliminated frequent sequences $|F(d)|$.

$$Score(d) = \frac{|V(d)|}{|F(d)|}. \tag{1}$$

$Score(d) = \infty$ in case $F(d) = 0$.

5.3 Border Representation

The remaining challenge is to efficiently compute $|V(d)|$ and $|F(d)|$. A naive approach is to first enumerate all possible violating and frequent sequences and then count the number of sequences containing d. Yet, Definitions 3.3.1 and 3.2 imply that the numbers of violating sequences and frequent sequences grow exponentially with respect to the number of distinct doublets. Therefore, this naive approach is not a feasible solution for a large dataset. We present a border-based approach to have the compressed representations of the notions. The compression is lossless. A similar approach was employed by [53] to represent itemsets, but we use borders to represent sequences in this paper.

Definition 5.3.1 (Anti-chain) A set of sequences S is an *anti-chain* if $\forall x, y \in S$, $x \npreceq y$ and $y \npreceq x$. ■

Definition 5.3.2 (Border) An upper bound UB and a lower bound LB form a *border*, denoted by $[UB, LB]$, if (i) both UB and LB are anti-chains, (ii) each element of UB is a subsequence of some element in LB, and (iii) each element of LB is a supersequence of some element in U. A border $[UB, LB]$ represents the set of sequences $\{z \mid \exists x \in U, \exists y \in L \text{ s.t. } x \preceq z \preceq y\}$. ■

To show that the set of violating sequences $V(T)$ and the set of frequent sequences $F(T)$ are representable by borders, we need to show that the borders are interval-closed.

Definition 5.3.3 (Interval-closed) A collection of sequences S is *interval-closed* if S contains all sequences $\{z \mid \forall x, y \in S, \forall z, x \preceq z \preceq y\}$. ■

Observation 5.3.1 *$V(T)$ and $F(T)$ are interval-closed.* ■

A violating sequence q is a *minimal violating sequence* (*MVS*) if every proper subsequence of q is not a violating sequence. The violating sequence and frequent sequence borders are defined as follows:

Definition 5.3.4 (Violating sequence (VS) border) The violating sequence (VS) border consists of an upper bound UB and a lower bound LB, where UB contains all minimal violating sequences and LB contains all maximal sequences y with support $|T(y)| \geq 1$. ■

Definition 5.3.5 (Frequent sequence (FS) border) The frequent sequence (FS) border consists of an upper bound UB and a lower bound LB, where UB contains all singleton doublets d with support $|T(d)| \geq max(K, K')$, and LB contains all maximal sequences y with support $|T(y)| \geq K'$. ■

The process of identifying a border $[UB, LB]$ for an interval-closed collection has been studied in [15]. The process of identifying minimal violating sequences has been studied in [35] and is explained in Section 5.1. A border $[UB, LB]$ can be represented by a set of *edges* $\{\langle x, y\rangle \mid x \in UB, y \in LB, x \preceq y\}$.

5.4 Counting Function

Suppose a doublet d has been suppressed in Line 4 in Algorithm 2. We need to efficiently compute the $Score(d')$ of the remaining doublets d' that share the same violating or frequent sequences with d. Specifically, $|V(d')|$ and $|F(d')|$ are decreased by the number of violating/frequent sequences containing both d and d' because such sequences have been eliminated. The question is how to compute such numbers from the borders without materializing the actual sequences. Equation 2 returns the number of sequences with maximum length L that are supersequences of a given sequence q and are covered by a border $[UB, LB]$ [53].

$$N_q^L([UB, LB]) = \{z \mid z \in [UB, LB], q \preceq z, |z| \leq L\}. \tag{2}$$

Consider a single edge $\langle x, y\rangle$ in a border. Equation 3 returns the number of sequences with maximum length L that are covered by $\langle x, y\rangle$ and are supersequences of a given sequence q.

$$\begin{aligned} N_q^L(\langle x, y\rangle) &= |\{z \mid x \preceq z \preceq y, q \preceq z, |z| \leq L\}| \\ &= \sum_{i=0}^{m} P(|y - (x \Cup q)|, i), \end{aligned} \tag{3}$$

where $\Cup$ unions two sequences and sorts the doublets by their timestamps, $P(n, i) = \frac{n!}{(n-i)!}$, and

$$m = min(|y - (x \Cup q)|, L - |x \Cup q|). \tag{4}$$

In the special case of $x = \emptyset$ and $L = \infty$, $N(\langle x, y\rangle) = 2^{|y-x|}$ returns the number of sequences covered by $\langle x, y\rangle$.

Example 5.4.1 Refer to the edge $\langle b4 \rightarrow f5, g2 \rightarrow a3 \rightarrow b4 \rightarrow f5\rangle$ in Figure 7. Suppose we want to suppress the sequence $q = \langle b4 \rightarrow f5\rangle$ with $L = 3$. Thus, we have $|x| = 2$, $|y| = 4$, $|q| = 2$, and $m = min(|g2 \rightarrow a3 \rightarrow b4 \rightarrow f5| - |(b4 \rightarrow f5 \Cup b4 \rightarrow f5)|, 3 - |(b4 \rightarrow f5 \Cup b4 \rightarrow f5)|) = 1$. The number of sequences removed due to the suppression is $N_q^L(\langle x, y\rangle) = \sum_{i=0}^{1} P(|y - (x \Cup q)|, i) = 3$, namely $\langle b4 \rightarrow f5\rangle$, $\langle a3 \rightarrow b4 \rightarrow f5\rangle$, and $\langle g2 \rightarrow b4 \rightarrow f5\rangle$. ■

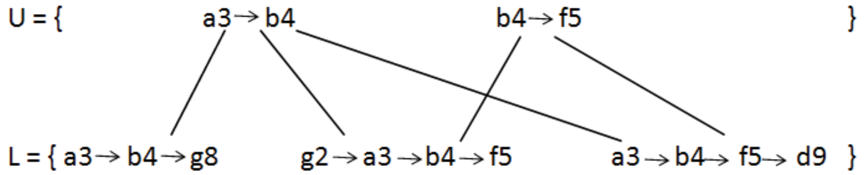


Figure 7: Violating Sequence Border

However, computing $N_q^L([UB, LB])$ by simply summing up $N_q^L(\langle x, y\rangle)$ over all edges $\langle x, y\rangle$ in $[UB, LB]$ is incorrect because a sequence may be covered by multiple edges, and we cannot count the same sequence more than once. Xu et al. [53] introduced two operations, *edge intersection* and *edge difference*, to remove duplicate counting.

5.5 Border-based Suppression Algorithm

We now present our border-based suppression algorithm. Initially, E is the set of all edges $\langle x, y\rangle$ in the border. d is the doublet to be suppressed with $d \in y$. d' denotes any other doublet in the suppressed sequence. E^* is the set of unexamined edges (initially E). $E^\wedge$ is the set of examined edges (initially empty). $Supp$ denotes the set of doublets to be suppressed. The algorithm iteratively suppresses sequences in E^* until all violating sequences are eliminated. It makes one pass of the edges in E^*. At each step, we consider the next edge $\langle x, y\rangle$ in E^*. Count the number of suppressed sequences, called losers, containing dd' that are covered by $\langle x, y\rangle$ but not covered by any (examined) edge in $E^\wedge$, and increment $\delta(d')$ by the count. Then move $\langle x, y\rangle$ from E^* to $E^\wedge$. This process is repeated until E^* becomes empty. The final $\delta(d')$ gives the number of losers containing dd'.

We summarize the main steps of Algorthim 2 as follows:

1. **Select the doublet to be suppressed:** Select doublet d not contained in $Supp$ with maximum Score. Add d to $Supp$.

2. **Get affected edges:** Retrieve $E(d) = \{\langle x, y\rangle, d \in y\}$. Set $E^* = E(d)$, $E^\wedge = \varnothing$

3. **Compute number of affected sequences:** The counting function returns the number of affected sequences (losers) by the current suppression to update the score. Consider Algorithm 3. Before it increases the score, it identifies the set of all edges in $E^\wedge$ that overlap with current edge $\langle x, y\rangle$, denoted by $ovset = \{e^\wedge | e^\wedge \in E^\wedge$ such that $\langle x, y\rangle \Cap e^\wedge \neq \emptyset\}$. To exclude the losers covered by ovset, consider in three cases:

   Case 1: $|ovset| = 0$. The losers covered by $\langle x, y\rangle$ are not covered by $E^\wedge$, so $N_q^L(\langle x, y\rangle)$ gives the number of new losers containing dd', where $q = dd'$ and $len = L$, the maximum length of the sequence. We update $\delta(d')$ to $\delta(d') + N_q^L(\langle x, y\rangle)$.

   Case 2: $|ovset| = 1$. In this case, only one edge in $E^\wedge$, say $e^\wedge$, has overlap with $\langle x, y\rangle$. The number of losers covered by both $\langle x, y\rangle$ and $e^\wedge$ is given by $N_q^L(\langle x, y\rangle \Cap e^\wedge)$, where $q = dd'$, $len = L$. We increment $\delta(d')$ by $N_q^L(\langle x, y\rangle - N_q^L(\langle x, y\rangle \Cap e^\wedge)$, where $\Cap$ is the intersection of two edges.

   Case 3: $|ovset| > 1$. In this case, more than one edge in $E^\wedge$ has overlap with $\langle x, y\rangle$. Simply excluding the intersections $\langle x, y\rangle \Cap e^\wedge$ for every $e^\wedge$ in $ovset$ does not work because intersections themselves might have intersection(s). Therefore, we pick any $e^\wedge$ in $ovset$ and compute $\langle x, y\rangle \Cap e^\wedge$. This edge difference can be replaced with a set of new edges denoted by $newset$. Then we recursively count the losers covered by the unexamined $E^* = newset$ but not by the examined $E^\wedge = ovset - e^\wedge$. The recursion terminates in either Case 1 or Case

Algorithm 3 Suppressing a Sequence

```
Procedure Compute $\delta(newset, ovset - \{e^\wedge\}, v, len, \delta)$;
while E* is not empty do
    Pick any edge $e^* = \langle x, y \rangle$ from $E^*$;
    let ovset = edges in $E^\wedge \cap e^*$;
    if $|ovset| = 0$ then /*case 1*/ then
        $\delta(d‘) = \delta(d') + N_q^L(e^*)$, for $d' \in y - d$;
    else if $|ovset| = 1$ then /*case 2*/ then
        let $e^\wedge$ be the edge in ovset
        $\delta(d') = N_q^L(*) - N_q^L(e^* \cap e^\wedge)$ , for $d' \in y - d$;
    else if $|ovset| > 1$ then /*case 3*/ then
        pick any edge $e^\wedge$ from ovset;
        set newest = $e^* - e^\wedge$ ;
        Compute$\delta(newset, ovset - \{e^\wedge\}, v, len, \delta)$;
    end if
    move E* from E* to $E^\wedge$;
end while
```

2.

4. **Update Score:** For every doublet in d', decrease $|VS(d')|$ by $\delta(d')$.

5. **Update the border**: This step removes all violating sequences containing d from VS border. For each $\langle x, y \rangle$ in $E(d)$, if $d \in x$, delete x from the upper bound of the border and delete all attached edges; if d $\notin x$, replace y by y'=y-d, delete y' if $y' \subseteq y''$ for some y'' on the lower bound.

6. **Repeat Steps 2-5** for the border of frequent sequences.

Chapter 6

Empirical Study

The main objective of our empirical study is to evaluate the performance of our proposed architecture and ST-Anonymizer in terms of *utility loss* caused by anonymization and *scalability* for handling large datasets. The utility loss is defined as $\frac{|F(T)|-|F(T)'|}{|F(T)|}$, where $|F(T)|$ and $|F(T)'|$ are the numbers of frequent sequences before and after the anonymization of the dataset T. We converted the data into relational data and attempted to apply the state-of-the-art anonymization algorithms, such as [22] [31] [46]. Unfortunately, all of these methods were not scalable to high dimensionality and failed to finish the anonymization.

We conducted the experiments on the *Metro100K* dataset, which simulates the travel routes of 100,000 passengers in the Montreal subway transit system with 65 stations in 60 minutes, forming 3,900 dimensions. Each record in the dataset corresponds to the route of one passenger. The passengers' traffic patterns are simulated based on information obtained from the Montreal metro information website[1]. Based on the published annual report, all of the passengers have an average

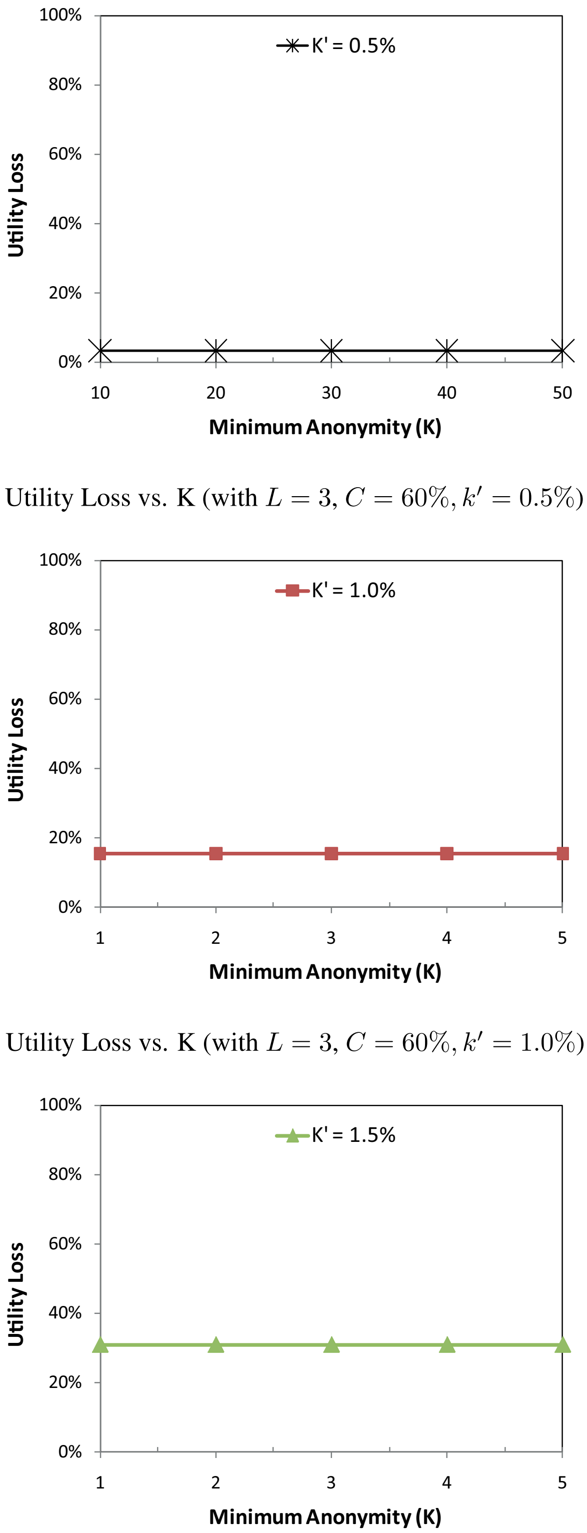
[1] http://www.metrodemontreal.com

spatial-temporal path length of 8 stations. The data generator also simulates the paths according to the current metro map and passengers' flow in each station. Each record contains an attribute with five possible values, one of which is considered to be sensitive.

6.1 Utility Loss

Following the convention for extracting frequent sequences, we specify the minimum support threshold at $K' = 0.5\%$, $K' = 1.0\%$, and 1.5% and vary the thresholds of minimum anonymity K, maximum confidence C, and maximum adversary's knowledge L to evaluate the performance of the algorithm.

Figure 8 depicts the utility loss for K from 10 to 50 while fixing $L = 3$ and $C = 60\%$. The utility loss stays flat with respect to the increase of K. As the K' increases from 0.5% to 1.5%, the number of frequent sequences decreases and the utility loss increases from 3% to 31% because a global suppression on a doublet generates a larger impact. Figure 9 depicts the utility loss for C from 20% to 100% while fixing $L = 3$ and $K = 30$. Approximately one-fifth of the records contain a sensitive value, so the utility loss is high at $C = 20\%$. As C increases, the effect of attribute linkages becomes insignificant. As K' increases, the utility loss drops quickly due to less overlapping between $F(T)$ and $V(T)$. Figure 10 depicts the utility loss for L from 1 to 9 while fixing $K = 30$ and $C = 60\%$. As L increases, the LKC-privacy requirement becomes harder to achieve and, therefore, requires more suppressions, resulting in higher utility loss.

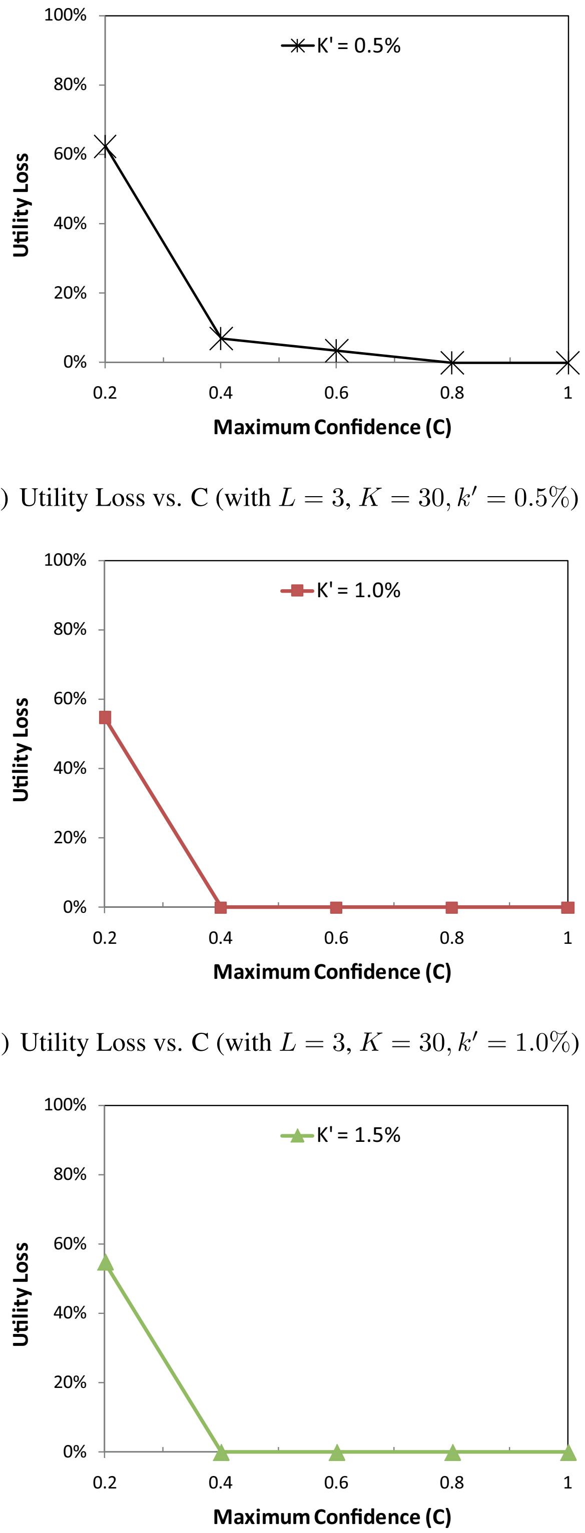


(a) Utility Loss vs. K (with $L = 3$, $C = 60\%, k' = 0.5\%$)

(b) Utility Loss vs. K (with $L = 3$, $C = 60\%, k' = 1.0\%$)

(c) Utility Loss vs. K (with $L = 3$, $C = 60\%, k' = 1.5\%$)

Figure 8: Utility Loss vs. K

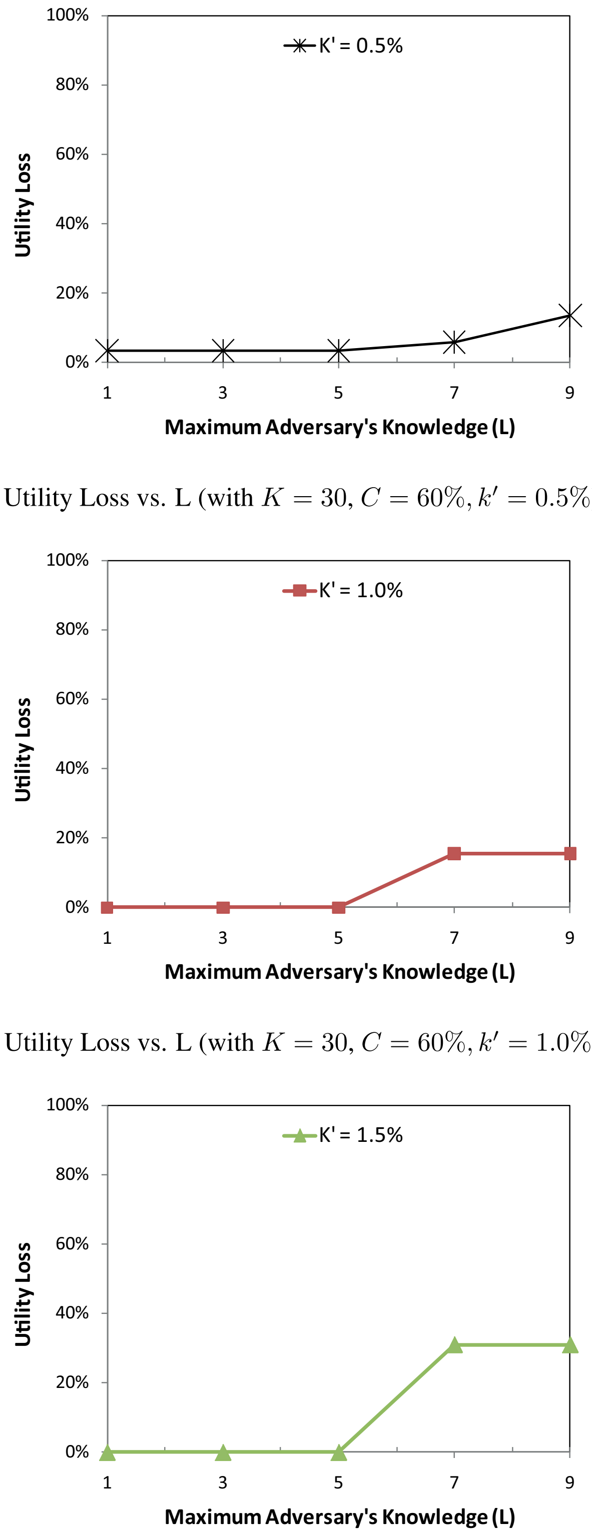


(a) Utility Loss vs. C (with $L = 3$, $K = 30, k' = 0.5\%$)

(b) Utility Loss vs. C (with $L = 3$, $K = 30, k' = 1.0\%$)

(c) Utility Loss vs. C (with $L = 3$, $K = 30, k' = 1.5\%$)

Figure 9: Utility Loss vs. C



(a) Utility Loss vs. L (with $K = 30$, $C = 60\%, k' = 0.5\%$)

(b) Utility Loss vs. L (with $K = 30$, $C = 60\%, k' = 1.0\%$)

(c) Utility Loss vs. L (with $K = 30$, $C = 60\%, k' = 1.5\%$)

Figure 10: Utility Loss vs. L

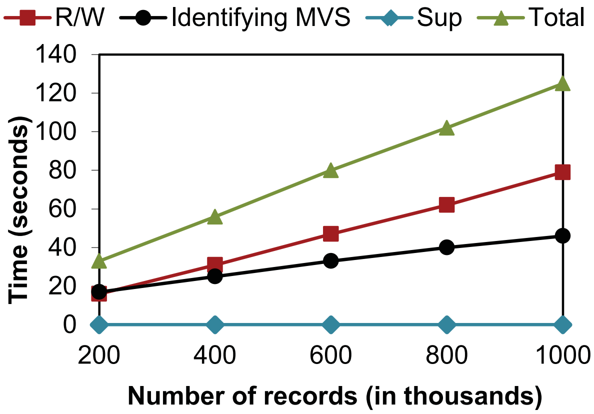


Figure 11: Runtime vs. number of records

6.2 Scalability

Every previous test case can finish the entire anonymization process within 15 seconds. We further evaluate scalability with respect to data volume and dimensionality. We fix $L = 3$, $K = 30$, $C = 60\%$, and $K' = 1\%$. Figure 11 depicts the runtime in seconds of from 200,000 to 1 million records. The total runtime for anonymizing 1 million records is 125 seconds, of which 46 seconds are spent identifying minimal violating sequences (MVS) and 79 seconds are spent reading the raw dataset and writing the anonymous dataset. It takes less than 1 second to suppress all the violating sequences $V(T)$. As the number of records increases from 200,000 towards 1 million, the runtime for read/write and identifying MVS also increases linearly, suggesting that our algorithm is scalable to anonymize large datasets. This high performance is due to the efficiency of computing the number of covered sequences by edges instead of enumerating such sequences. This eliminates the need to store all frequent sequences and violating sequences in memory, which is the real bottleneck due to the exponential blowup of $|F(T)|$ and $V(T)|$. All experiments are conducted on a PC with Intel Core2 Duo 1.6GHz CPU with 2GB of RAM.

Chapter 7

Conclusion and Future Work

We have studied the problem of privacy-preserving spatial-temporal data sharing and have proposed a service-oriented architecture together with an anonymization algorithm to simultaneously preserve both privacy and information utility for data mining. Applying K-anonymity on high-dimensional data, such as the spatial-temporal data in our experiments, would result in a high utility loss. To overcome the problem, we adopt a LKC-privacy model based on a practical assumption that an adversary has limited background knowledge about the victim. Furthermore, we propose a border-based anonymization method to compress the large number of violating and frequent sequences into a compact format to ensure the scalability of the system.

For our future work, we would like to develop a secure protocol to integrate distributed spatial-temporal data owned by different location-based service providers, such that the integrated data satisfies a privacy model such as LKC-privacy.

Bibliography

[1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 24th IEEE International Conference on Data Engineering*, pages 376–385, Cancun, Mexico, April 2008.

[2] O. Abul, F. Bonchi, and M. Nanni. Time series compressibility and privacy. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB)*, pages 459–470, 2008.

[3] C. Aggarwal and P. S. Yu. On privacy-preservation of text and sparse binary data with sketches. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2007.

[4] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the VLDB*, 2005.

[5] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th Very Large Data Bases (VLDB)*, pages 487–499, 1994.

[6] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Computer System Science 58*, pages 137–147, 1999.

[7] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proceedings of the 21st IEEE International Conference on Data Engineering (ICDE)*, 2005.

[8] Z. Berenyi and H. Charaf. Retrieving frequent walks from tracking data in RFID-equipped warehouses. In *Proceeding of the Human System Interactions*, 2008.

[9] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 1:46–55, 2003.

[10] A. J. Brimicombe. Gis - where are the frontiers now? In *Proceedings of GIS*, 2002.

[11] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Comm. ACM 24,2*, pages 84–88, 1981.

[12] R. Chen, B. C. M. Fung, N. Mohammed, and B. C. Desai. Privacy-preserving trajectory data publishing by local suppression. *Information Sciences: Special Issue on Data Mining for Information Security*, in press.

[13] L. H. Cox. Suppression methodology and statistical disclosure control. . *Am. Statistical Assoc. 75, 370*, pages 377–385, 1980.

[14] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the ACM PODS*, 2003.

[15] G Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of 5th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 1999.

[16] C. Dwork. Differential privacy. In *Proceedings of the ICALP*, 2006.

[17] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of 3rd Theory of Cryptography Conference*, pages 265–284, New York, USA, March 2006.

[18] B. C. M. Fung, K. Al-Hussaeni, and M. Cao. Preserving RFID data privacy. In *Proceedings of the 2009 International Conference on RFID*, pages 200–207, Orlando, FL, April 2009. IEEE Communications Society.

[19] B. C. M. Fung, M. Cao, B. C. Desai, and H. Xu. Privacy protection for RFID data. In *Proceedings of the 24th ACM SIGAPP Symposium on Applied Computing (SAC)*, pages 1528–1535, Honolulu, HI, March 2009.

[20] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (CSUR)*, 42(4):1–53, December 2010.

[21] B. C. M. Fung, K. Wang, and P. S. Yu. In *Top-down specialization for information and privacy preservation*, pages 205–216, 2005.

[22] B. C. M. Fung, K. Wang, and P. S. Yu. Anonymizing classification data for privacy preservation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(5):711–725, May 2007.

[23] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 2007.

[24] G. Ghinita, Y. Tao, and P. Kalnis. On the anonymization of sparse high-dimensional data. In *Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE)*, pages 715–724, April 2008.

[25] H. Gonzalez, J. Han, and X. Li. Mining compressed commodity workflows from massive RFID data sets. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM)*, 2006.

[26] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the MobiSys*, 2003.

[27] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *Proceedings of the ACM CCS*, 2007.

[28] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the 8th ACM SIGKDD*, pages 279–288, 2002.

[29] M. Jakobsson, A. Juels, and R. L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In *Proceedings of the 11th USENIX Security Symposium*, pages 339–353, 2002.

[30] A. Juels. RFID security and privacy: a research survey.

[31] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of ACM International Conference on Management of Data (SIGMOD)*, pages 49–60, Baltimore, ML, 2005.

[32] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proceedings of the IEEE ICDE*, 2006.

[33] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, March 2007.

[34] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proceedings of the 23rd ACM SIGMOD-SIGACT-SIGART PODS. ACM, New York*, pages 223–228, 2004.

[35] N. Mohammed, B. C. M. Fung, and M. Debbabi. Walking in the crowd: Anonymizing trajectory data for pattern analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 1441–1444, Hong Kong, November 2009.

[36] 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 International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1285–1294, June 2009.

[37] R. G. Pensa, A. Monreale, F. Pinelli, and D. Pedreschi. Pattern-preserving k-anonymization of sequences and its application to mobility data mining. In *Proceedings of the International Workshop on Privacy in Location-Based Applications*, 2008.

[38] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(6):1010–1027, 2001.

[39] N. Shiode, C. Li, M. Batty, P. Longley, and D. Maguire. The impact and penetration of location based servicess. 2004.

[40] S. Steinger, M. Neun, and A. Edwards. Foundations of location based services, 2004.

[41] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(5):571–588, 2002.

[42] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *Proceedings of the 9th International Conference on Mobile Data Management*, pages 65–72, Beijing, China, April 2008.

[43] M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving anonymization of set-valued data. *Proceedings of the VLDB Endowment*, 1(1):115–125, August 2008.

[44] K. Virrantaus, J. Markkula, A. Garmash, and Y. V. Terziyan. Developing gis-supported location-based services. In *Proceedings of WGIS 2001, First International Workshop on Web Geographical Information Systems*, 2001.

[45] K. Wang, B. C. M. Fung, and P. S. Yu. Template-based privacy preservation in classification problems. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM)*, pages 466–473, 2005.

[46] K. Wang, B. C. M. Fung, and P. S. Yu. Handicapping attacker's confidence: An alternative to k-anonymization. *Knowledge and Information Systems (KAIS)*, 11(3):345–368, April 2007.

[47] S. Wang, J. Min, and K. Byung. Lg electronics mobile research. In *Proceedings of the IEEE ICC 2008 - Beijing*, 2008.

[48] S. L. Warner. Procceedigs of the randomized response: A survey technique for eliminating evasive answer bias. In *J. Am. Statistical Assoc. 60*, pages 63–69, 1965.

[49] A. R. C. Wong, J. Li, A. W. C. Fu, and K. Wang. (alpha,k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the 12th ACM SIGKDD. ACM, New York*, pages 754–759, 2006.

[50] R. C. W. Wong, J. Li., A. W. C. Fu, and K. Wang. (α,k)-anonymous data publishing. *Journal of Intelligent Information Systems*, 33(2):209–234, October 2009.

[51] X. Xiao and Y. Tao. Personalized privacy preservation. *SIGMOD*, 2006.

[52] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W. C. Fu. Utility-based anonymization using local recoding. In *Proceedings of the 12th ACM SIGKDD Conference. ACM, New York*, 2006.

[53] Y. Xu, B. C. M. Fung, K. Wang, A. W. C. Fu, and J. Pei. Publishing sensitive transactions for itemset utility. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, December 2008.

[54] Y. Xu, K. Wang, A. W. C. Fu, and P. S. Yu. Anonymizing transaction databases for publication. In *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2008.

[55] Z. Yang, S. Zhong, and R. N. Wright. Anonymity-preserving data collection. In *Proceedings of the 11th ACM SIGKDD Conference. ACM*, pages 334–343, 2005.

[56] R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, and W. H. Wang. Anonymizing moving objects: How to hide a MOB in a crowd? In *Proceedings of the 12th International Conference on Extending Database Technology (EDBT)*, pages 72–83, Saint-Petersburg, Russia, March 2009.

[57] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*, 2007.