Towards a Privacy Preserving Framework for Publishing Longitudinal Data

by

Morvarid Sehatkar

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements for the degree of

Ph.D. in Computer Science

Under the auspices of the Ottawa-Carleton Institute for Computer Science

University of Ottawa

Ottawa, Ontario, Canada

Abstract

Recent advances in information technology have enabled public organizations and corporations to collect and store huge amounts of individuals' data in data repositories. Such data are powerful sources of information about an individual's life such as interests, activities, and finances. Corporations can employ data mining and knowledge discovery techniques to extract useful knowledge and interesting patterns from large repositories of individuals' data. The extracted knowledge can be exploited to improve strategic decision making, enhance business performance, and improve services. However, person-specific data often contain sensitive information about individuals and publishing such data poses potential privacy risks. To deal with these privacy issues, data must be anonymized so that no sensitive information about individuals can be disclosed from published data while distortion is minimized to ensure usefulness of data in practice. In this thesis, we address privacy concerns in publishing *longitudinal* data. A data set is longitudinal if it contains information of the same observation or event about individuals collected at several points in time. For instance, the data set of multiple visits of patients of a hospital over a period of time is longitudinal. Due to temporal correlations among the events of each record, potential background knowledge of adversaries about an individual in the context of longitudinal data has specific characteristics. None of the previous anonymization techniques can effectively protect longitudinal data against an adversary with such knowledge. In this thesis we identify the potential privacy threats on longitudinal data and propose a novel framework of anonymization algorithms in a way that protects individuals' privacy against both identity disclosure and attribute disclosure, and preserves data utility. Particularly, we propose two privacy models: $(K, C)^P$-*privacy* and (K, C)-*privacy*, and for each of these models we propose efficient algorithms for anonymizing longitudinal data. An extensive experimental study demonstrates that our proposed framework can effectively and efficiently anonymize longitudinal data.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr Stan Matwin, for his constant guidance and support throughout my PhD. It was an honor for me to be his student. I am also thankful to Dr Khaled El Emam, at the Electronic Health Information Laboratory at the CHEO Research Institute, for his inspiration with this research.

Many thanks to my family, specially my parents for their unconditional support and love, and my brother, Mohammad, who encouraged me and supported me to start this journey.

Finally, I owe a very special thanks to my husband, Amir Sadeghian, for his patience, encouragement, and understanding. Without his support this thesis would remain incomplete.

To my husband Amir, Mom and Dad ...

Contents

List of Tables

List of Figures

Chapter 1

Introduction

"Nothing is your own except the few cubic centimeters inside your skull". This statement from Orwell's novel, *1984* [88], is no longer fiction and has become a reality today. On the one hand, recent advances in information technology have enabled public organizations and corporations to collect and store huge amounts of individuals data, including credit history, medical records, purchase history, criminal records, web search logs, etc. in data repositories. On the other hand, the development of advanced data mining and knowledge discovery techniques has enabled government agencies and organizations to dig through huge collections of data and effectively extract useful knowledge and previously unknown patterns.

In other words, almost *everything* about an individual's life like credit cards transactions, purchases, emails, phone calls, web searches, flights and hotels bookings, doctor visits, crimes, accidents, visited places, etc, are being stored in databases that can be subsequently utilized and analyzed by government departments and corporations to extract patterns about characteristics and behaviors of individuals, even those things that individuals may not know about themselves.

The extracted knowledge can be exploited by organizations and government agencies in a wide range of public and private sector activities to improve their strategic decision making and better serve their customers and the public. Governments can employ such

knowledge to evaluate public programs and investments, for instance in crime, public health, economic growth, social security, and law enforcement to improve public services [112]. Funding projects such as TIA(Total Information Awareness)[1] and MATRIX(The Multistate Anti-TeRrorism Information eXchange)[2], by the US federal government, are two examples of data mining applications proposed to improve national security by identifying potential suspects related to terrorist attacks. Companies in public and private sector, particularly with a consumer focus, such as retail, financial, and marketing organizations can extract the useful trends from their customers purchase data or transactions to improve their target marketing, customer satisfaction, services, and profits. A financial institution can analyze the monthly transactions of its cardholders to suggest customized products to them. A retailer can use records of customer purchases to identify customer buying patterns and offer targeted promotions based on an individual's purchase history. For example, *Amazon.com* employs its stored information about the items that were viewed, bought, and sold by every customer to extract patterns for improving its customized services [8]. Health care organizations can use medical data to discover health and medical related patterns in order to enhance the quality of health care and improve public health. For instance, Partners HealthCare[3], which is a nonprofit integrated health care system, is using its health data in order to discover the patterns related to drug usage and clinical events [93].

Considering the numerous benefits of storing and analyzing person-specific data, demand for collecting, sharing and exchanging such data among various organizations and even making data publicly available has been rapidly increasing. However, in most cases person-specific data contain sensitive information and publishing such data may violate the privacy of individuals. For example, in medical data, the information about a patient's disease, such as HIV, may be considered sensitive. In trajectory data, a certain location visited by an individual may be sensitive. In web search data, some specific

[1] http://en.wikipedia.org/wiki/Information_Awareness_Office

[2] http://en.wikipedia.org/wiki/Multistate_Anti-Terrorism_Information_Exchange

[3] http://www.partners.org/

websites which are surfed by a user may be considered as sensitive information [127]. This fact has raised major privacy concerns among the public about misuse of their data and has led to one of the highly debated ethical issues related to technology: data mining vs individual privacy. The majority of these concerns are related to the *secondary use of personal information* where the collected data are used for purposes other than those for which the data were originally collected [14]; For instance when personal health information data are used for research, public health, or marketing [98]. Such concerns are mostly due to the fact that individuals, whose data are stored in data repositories, often are not aware of what is happening to their data "behind the scenes"[51]. Therefore their concerns have been highly raised about the misuse of their data [96].

Actually, such concerns are not unfounded and several real cases of privacy violations through accessing to individuals' data have been recently reported. For instance, in 2006 two New York Times reporters could re-identify a single 62 year old woman living in Lilburn, Georgia through her specific search queries in a published AOL search log dataset [11]. Another example is for the case of Netflix [4] data which was published for a competition to improve the accuracy of Netflix movie recommender system, but it was found that 96% of users can be uniquely identified by just using the information of at most 8 movie ratings in the published data [127]. Sweeny [111] assessed a public real medical data and could identify the medical record of a former governor of Massachusetts by matching his record with a publicly available voter registration list. All these cases demonstrate the importance of protecting individuals' data to prevent serious consequences that can happen through sensitive information disclosure.

A naive solution to address such concerns is to *not* release data [127]. But this will prevent the governments, organizations, and society from taking advantage of analyzing individuals' data to identify interesting trends or patterns. One may also think that instead of publishing data, it would be better that data owners release the statistical data or the result of data mining. However, it should be noted that data owners typically do

[4]https://www.netflix.com/

not have any data mining expertise and just want to simply release data to data recipients or the public. Moreover, having access to data gives data recipients more flexibility to perform different types of analysis on data. Obviously, it would not be practical that for any new data analysis task, data recipients make a request to data owners to generate new statistical or data mining results. Also, sometimes the data must be publicly available in accordance with the law. For example, to comply with the privacy regulations, all hospitals in California must publish their patients datasets on the web [19]. Thus, simply not publishing data is not a good solution.

To address these challenges, governments and ethics boards regulated a set of privacy policies to control and restrict the publication and usage of individuals' data. For instance, US congress enacted $HIPAA$ (the Health Insurance Portability and Accountability Act)[5] which regulates the use and disclosure of health information across the US. Equivalently, $PHIPA$ (the Personal Health Information Protection Act)[6] was legislated in Ontario, Canada. These privacy regulations aim to address privacy concerns either by imposing restrictions on accessing sensitive data or by enforcing researchers and organizations to obtain the consent of data owners for using their data. Moreover, these regulations provide some guidelines for organizations to acquire agreements from researchers, before releasing data, for the right usage of data. However, such requirements adversely impact the potential advancements of usage and analysis of individuals data [14]. For instance, obtaining consent from each patient for using his/her personal health information for research purposes is often impossible in secondary use contexts. Moreover, those laws might not be able to effectively protect individuals' privacy. For example, signing agreements by researchers can not completely guarantee that the data will not mistakenly be accessed by a third party which is not supposed to have such access [41].

An alternative and more promising approach to protect the privacy of individuals is

[5]http://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act

[6]http://en.wikipedia.org/wiki/Personal_Information_Protection_and_Electronic_Documents_Act

to anonymize data before publishing [41, 124]. For this purpose, the data publisher must modify data such that no sensitive information about individuals can be disclosed from published data, while the data remain useful for analysis. More precisely, the difference in the performance of analyzing the anonymized data and the original data should not be significant. This is a challenging problem, since any modification of the data distorts data utility. So, the data publisher must find the modification procedure that preserves the maximum data utility. A conventional approach to data anonymization has been to remove explicit identifiers of individuals, such as names or social security numbers, from datasets. However, as Sweeney showed [111], simply removing explicit identifiers is not sufficient to protect the privacy of individuals. She demonstrated that some non-identifying attributes, like date of birth, race, ZIP code or gender, called *quasi-identifiers*, can be linked to publicly available databases, e.g. a voters database, to disclose personal information. Therefore, more effective methods are required for data anonymization to prevent privacy attacks. The goal of *privacy preserving data publishing* [41] is to address this challenge by providing methods and tools for publishing data such that the released data protects the privacy of individuals, while the utility of resulting data is preserved for data analysis tasks. Figure 1.1 shows a privacy preserving data publishing scenario. As can be seen in Figure 1.1, in every data publishing scenario usually there are three parts including data publisher, data recipient, and adversary. A data publisher is an organization, such as a hospital, whose goal is to publish a person-specific data to data recipients such that privacy of individuals will not be violated. A data recipient is an organization, a research group, or a person whose goal is to use data for some data mining and data analysis tasks. An adversary is a person or a group whose goal is to attack a published person-specific data to obtain some new sensitive information about individuals. An adversary may have some background knowledge or employ some publicly available data sources to launch a privacy attack.

1.1 Motivation

There are various types of data that can be published for data mining and data analysis tasks, including relational data, transaction data, sequence data, trajectory data, and graph data. Each category contains datasets with different information which are being widely used in data mining applications, such as astrology, weather forecasting, agriculture, marine, etc. However, privacy concerns raise when some information about individuals are being published. Thus in this research we only consider person-specific data and everywhere we talk about data, we mean person-specific data.

Various data types differ in structure, properties and the information they contain about individuals. Publishing data of any specific type has its own privacy risks with respect to potential background knowledge that an adversary can obtain in that context. For instance, assume that a retailer, such as a grocery store, wants to publish its customer purchase transaction data. An adversary may have seen a subset of the items that a target individual bought from this grocery store and she employs such background knowledge to re-identify transaction of that individual in the published transaction dataset. As another example, an adversary who knows a subset of the places that a target individ-

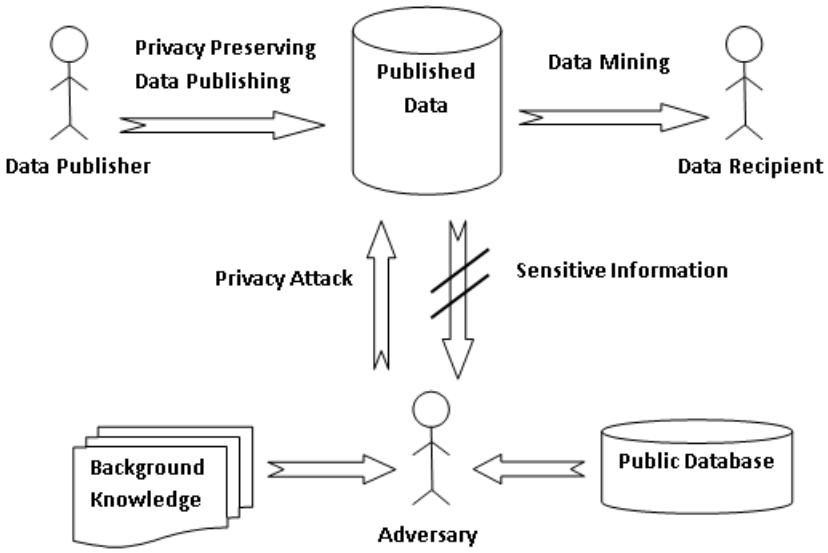


Figure 1.1: Privacy preserving data publishing scenario

ual has visited at specific times, can launch a privacy attack on a published trajectory dataset, containing the information of that individual, to infer some sensitive information. Therefore, for each type of data a specific anonymization technique is required in order to effectively protect data against potential privacy attacks specific to that data type.

Clinical data found in EMRs and EHRs contain information of multiple visits of patients and therefore these data are longitudinal by nature. A dataset is longitudinal if it contains information of the same observation or event about an individual collected at several points in time. For instance, the information which is collected in clinical trials to evaluate the impact of treatments on a disease over time constitutes longitudinal data [7].

Today, with advances in health informatics, the majority of health data are stored in Electronic Health Records (EHR) and Electronic Medical Records (EMR) at medical facilities. This means that a huge amount of Personal Health Information(PHI) is available that can be shared for secondary purposes, such as research and marketing. Secondary use of PHI has substantial advantages such as enhancing the quality of health care, expanding knowledge about diseases and treatments, providing better understanding about effectiveness and efficiency of health care systems, supporting public health and security goals, and helping businesses to meet customers' needs [98]. However, personal health information in comparison with other personal data is more private and sensitive, and improper release and usage of person-specific health data can cause critical harm such as the discrimination, stigmatization, or loss of insurance or employment [55]. For example, disclosure of sensitive information of a patient who is diagnosed with bipolar disorder or HIV may lead him to lose a potential job opportunity. Therefore, health data should be anonymized to address the privacy issues in publishing personal health information.

Longitudinal data resides in the category of sequence data. More precisely, it refers to a sequence data containing multi-dimensional event sequences [30]. Other examples

of longitudinal data are clickstream data and customers purchase histories [30]. The term "longitudinal data" have been widely used in health care domain to refer to those datasets that contain sequences of clinical events related to patients. An example of longitudinal health data is shown in Table 1.1. Every record in the table corresponds to one visit of a patient and contains a fixed number n of attributes. More precisely, every record in this dataset has attributes PID (i.e., patient ID), VID (i.e., visit ID), $AdmYr$ (i.e., admission year), ZIP (ZIP code), $DSFC$ (i.e., number of days since first claim in each year), LOS (i.e length of stay in the hospital), and one sensitive attribute $Disease$. Each patient may have multiple records corresponding to her multiple visits in this dataset and visits of a patient are ordered with respect to visit ID which is assigned based on the visit date. For instance, from data in table 1.1, we find that there was a patient who had a visit in 2009 while he was living in ZIP code 56942 and *later* in another visit in 2010 he was hospitalized for 30 days, i.e., $PID = 3$.

Over the past years, several techniques have been proposed for privacy preserving publication of different types of data [41] but, to the best of our knowledge, no satisfactory approach has been proposed so far to address privacy issues of publishing *longitudinal data*. Potential background knowledge of adversaries about an individual in the context of longitudinal data has its own characteristics and is different from such knowledge in the context of other types of data. This is a challenge in anonymizing longitudinal data which cannot be effectively addressed by previous anonymization techniques. For example, consider the longitudinal health data in Table 1.1. An adversary may know the exact number of visits of a patient or she may just have information about *some* visits of a patient. It is also possible that the adversary knows just the values of some of the attributes in some visits. A more complex case is when the adversary knows the time difference between 2 or more visits of a patient. For example, the adversary may hear from her neighbor that he has been recently in the hospital and his doctor gave him an MRI and his appointment is 40 days later. He may complain that his appointment is too late since he is very concerned about his disease. This type of conversation is

PID	VID	AdmYr	ZIP	DSFC	LOS	Disease
1	1	2009	56117	0	3	Hepatitis
2	1	2009	56103	0	2	Infection
3	1	2009	56942	0	1	Fever
3	2	2010	56942	0	30	Cancer
4	1	2009	56107	0	2	Cancer
4	2	2010	56107	0	35	Flu
5	1	2009	56117	0	3	Fever
6	1	2008	56103	0	3	Flu
6	2	2009	56103	0	1	Fever
6	3	2009	56230	40	2	HIV
7	1	2008	56072	0	2	Flu
8	1	2007	56361	0	30	Hepatitis
8	2	2011	56107	0	3	HIV
9	1	2007	56230	0	35	Flu
9	2	2011	56107	0	3	HIV
10	1	2009	56072	0	2	Flu
10	2	2010	56103	0	35	Fever
10	3	2010	56043	100	30	Infection

Table 1.1: Inpatient Longitudinal Data

common among people. If later, the adversary gets access to the longitudinal data of that hospital, remembering that her neighbor was complaining about the long waiting time for MRI, she can launch a privacy attack on the data with the goal of re-identifying the record of her neighbor. If there are only one or a few patients with two consecutive visits that occurred in 40 days, the adversary may be able to re-identify the record of her neighbor. This specific type of knowledge that an adversary may obtain makes the problem of longitudinal data anonymization more challenging and complicated. None of the previous anonymization techniques can effectively protect data against an adversary with such knowledge.

In general, an adversary who has some background knowledge about visits of a target individual is able to launch two kinds of privacy attacks on a published longitudinal data: *Identity disclosure* and *attribute disclosure*. In the former, if the adversary can find one or a few matches for her background knowledge in the dataset, then she will

be able to uniquely (or nearly uniquely) identify the record of the target individual. In the case of attribute disclosure, the adversary will be able to infer sensitive value(s) of a target individual without identifying an individual's record if all or most of the matching records to her background knowledge have the same sensitive value(s). Here are some examples of identity disclosure and attribute disclosure attacks which an adversary can launch on longitudinal data in Table 1.1:

- If the adversary knows that Bob had a visit in 2008 and he has been living in ZIP code 56230 from 2009, then she can uniquely identify Bob's record, #6, and consequently conclude that *Bob* has HIV.

- If the adversary knows that Bob had a visit in 2007 and later in 2011 he was hospitalized for 3 days, then both records #8 and #9 match her background knowledge. Although, she cannot uniquely identify Bob's record, she can conclude that Bob has HIV with 100% confidence as both records have sensitive value HIV in one of their visits.

- If the adversary knows that Bob had an MRI appointment in the hospital 40 days after one of his visits in 2009, then she can uniquely identify Bob's record, #6, and infer that he has HIV. This is the case of having background knowledge about the time difference between visits of a patient, discussed above.

In order to prevent these privacy attacks, longitudinal data should be anonymized such that no combination of values of quasi-identifiers within an event and across events of any record leads to privacy breach. Moreover, the applied anonymization method must consider temporal correlation among events in each record; otherwise it cannot provide adequate privacy protection.

There are a number of interesting research questions about publishing longitudinal data that need to be answered:

- What are the possible forms of background knowledge of an adversary in real life longitudinal data publishing scenarios?

- What are the potential privacy attacks that can be launched by an adversary?
- What are the challenges in designing an effective algorithm for longitudinal data anonymization?
- What is the best anonymization solution for longitudinal data which does not significantly compromise data utility?

This thesis aims to answer these questions.

1.2 Contributions

In this thesis, we study the problem of privacy preserving publication of longitudinal data. We identify the potential privacy threats on longitudinal data and propose a novel framework of anonymization algorithms to remove the identified privacy threats. We mainly focus on longitudinal health data and design anonymization techniques with respect to specific requirements of publishing health data. However, the techniques that we propose will be applicable to any multi-dimensional event sequence data. Our goal is to develop anonymization techniques to prevent both identity disclosure and attribute disclosure attacks. This is the first work to anonymize longitudinal data containing multi-dimensional events to prevent both identity disclosure and attribute disclosure. Below are the key contributions of this thesis:

- We survey the existing privacy preserving data publishing approaches in the literature and we propose a taxonomy including those approaches.
- In our first proposed approach, we assume an adversary who knows values of at most P quasi-identifiers of a target individual as well as the order of these quasi-identifiers in the records of the individual in longitudinal data. Based on this assumption, we formally define a new privacy notion for longitudinal data, called $(K, C)^P$-*privacy*, to prevent both identity disclosure and attribute disclosure. We

develop an efficient hybrid anonymization algorithm, called $HALT$, using global generalization and global suppression to effectively preserve privacy and utility in longitudinal data. This is the first work which employs the notion of generalized itemsets and generalized sequences, which come from frequent pattern mining, for data anonymization. Our experiments on different datasets demonstrate the effectiveness of our proposed approach for longitudinal data anonymization.

- In our second approach, we do not bound the maximum knowledge of an adversary and assume that the adversary may know any number of quasi-identifiers. This assumption is due to the fact that there are some scenarios where it may not be possible to determine the maximum knowledge of the adversary in advance. We define a privacy model called (K,C)-*privacy* to anonymize longitudinal data to prevent identity disclosure and attribute disclosure. We achieve (K,C)-privacy by iteratively clustering records using agglomerative hierarchical clustering [53] and sequence alignment [84] techniques. Using hierarchical clustering we group similar records together where similarity is determined based on our sequence alignment method. We apply generalization and suppression to each cluster, containing the most similar records, separately. The proposed framework is able to generate anonymized data with low information loss in an efficient manner.

- Due to privacy concerns, access to person-specific data is restricted. Lack of publicly available benchmark data has been always a challenge in evaluation and comparison of privacy preserving techniques in practice. This thesis takes an important step to resolve this problem by generating synthetic data based on two gold-standard health data sets, namely *CMS Inpatient Claims DE-SynPUF* data provided by the *Centers for Medicare and Medicaid Services* (CMS) [7] and *Heritage Health Prize* (HHP)[8] claims dataset. The results of our experiments confirm that

[7] http://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/SynPUFs/DE_Syn_PUF.html

[8] http://www.heritagehealthprize.com/c/hhp/data

both our algorithms can effectively anonymize data, while preserving data utility.

The results of this thesis have been published in [105, 104, 38, 103, 102], as listed below:

1. SEHATKAR, M. and MATWIN, S. Clustering-based multi-dimensional sequence data anonymization. In *International Workshop onPrivacyand Anonymity in the Information Society (PAIS)* (2014), pp. 385-389.[105]

2. SEHATKAR, M. and MATWIN, S. HALT: Hybrid anonymization of longitudinal transactions. In *Privacy, Security, Trust (PST)* (2013), pp. 127-134.[104]

3. EL EMAM, K., JONKER, E., SEHATKAR, M., WUNDERLICH, J. and GAUDETTE, L. Managing the risk of re-identification for public use files. *Office of the Privacy Commissioner of Canada* (2011).[38]

4. SEHATKAR, M. Privacy preserving publication of longitudinal health data. *In Canadian Conference on AI* (2010), pp. 412-413.[103]

5. SEHATKAR, M. Embedding anonymity protection in a concept lattice based association rule mining algorithm. In *The Third Workshop for Women in Machine Learning (WiML)* (2008).[102]

1.3 Thesis Organization

The rest of this thesis is organized as follows:

- In Chapter 2, we present the background concepts related to privacy preserving data publishing.

- In Chapter 3, we review the state-of-the-art in privacy preserving data publishing literature. We classify the existing techniques into two major categories based on the types of disclosure, i.e., identity disclosure and attribute disclosure. In each of these categories, we talk about the types of background knowledge of an

adversary as well as the potential privacy attacks based on such knowledge. Then, we discuss the proposed solutions in the literature to prevent the disclosure through those privacy attacks. We classify these methods into three categories based on the structure of the data being anonymized, including relational data, transaction data, and trajectory data.

- Chapter 4 presents a hybrid anonymization technique for longitudinal data using global generalization and global suppression to satisfy $(K, C)^P$-privacy.
- In Chapter 5, we present a clustering-based anonymization technique which uses agglomerative clustering and sequence alignment methods to anonymize longitudinal data while ensuring (K,C)-privacy model.
- Chapter 6 reports the experimental results of our proposed methods.
- Chapter 7 concludes the thesis and presents some directions for future work.

Chapter 2

Privacy Preserving Data Publishing: Background Concepts

As we discussed in Chapter 1, when a data set is being released for data mining purposes, a privacy preserving technique is required to prevent the disclosure of sensitive information of individuals. The goal of this chapter is to give a review of fundamentals in privacy preserving data publishing.

In a data publishing scenario, a data publisher, such as a hospital, releases its dataset to either a data miner, such as a researcher who uses the health data for medical research, or the public when such release is regulated by privacy laws. Since the dataset being released often contains sensitive information about individuals, such as the diseases of patients visiting a hospital, the data publisher can not release his data in a raw format due to its potential privacy risks.

In general, there are three types of disclosure that can lead to a privacy breach: *identity disclosure*, *attribute disclosure*, and *membership disclosure*. An identity disclosure, also known as *record linkage* [41], occurs when an adversary who has some background knowledge about a target individual, referred to as *victim*, can uniquely (or nearly uniquely) identify the record of that victim in a published data set, and, subsequently, infer his/her sensitive information. In an attribute disclosure, also known as *attribute*

linkage [41], the adversary is able to extract some sensitive information from data about the victim without identifying his/her record. In these two types of disclosure, the adversary knows that the record of the target individual exists in the published dataset and her goal is to obtain some sensitive information about that individual from that data. However, sometimes the presence or absence of an individual in a dataset is a sensitive piece of information. This leads to the third type of disclosure, i.e., membership disclosure, also known as *table linkage* [41], in which the adversary's goal is to find whether the victim's record is in a released dataset or not.

In order to effectively protect the privacy of individuals, the data publisher must ensure that these disclosures cannot occur in the released dataset. It has been shown that simply removing the explicit identifiers, such as names and social insurance numbers, cannot provide adequate protection [111] and more effective techniques are needed. As we discussed in chapter 1, a promising approach to protect the privacy of individuals is to anonymize data before publishing. Data anonymization is a privacy preserving data publishing approach that modifies data with the goal of hiding the identity and/or sensitive data of individuals. However, data modification leads to loss of information that destroys the utility of data. Therefore, the challenge in privacy preserving data publishing is to anonymize data such that the privacy of individuals is protected while data utility is preserved for analysis purposes.

In this chapter, we explain different aspects in privacy preserving data publishing, including types of datasets that are being published, anonymization techniques, and information loss and data utility measures.

2.1 Types of Data

There are various types of data among which relational data, transaction data, sequential data, trajectory data, and graph data have been mostly studied in privacy preserving data publishing research community. In the following we describe these data types.

- **Relational data:** this type of data has a fixed set of attributes which are common among a collection of records. A dataset of this type can be represented by a table in which each column corresponds to one attribute and each row represents one record. In general, four types of attributes may exist in a relational dataset [41]: *explicit identifiers*, *quasi-identifiers(QI)*, *sensitive attributes*, and *non-sensitive attributes*. The explicit identifiers are the attributes that uniquely identify individuals such as "name" and "social security number" and are always removed before releasing data. Quasi-identifiers are those attributes that are not identifier attributes but can potentially identify an individual especially when grouped together, such as "date of birth" and "postal code". Sensitive attributes are the sensitive information of individuals, such as "disease" and "salary". Any attribute that is not an explicit identifier, quasi-identifier or sensitive attribute resides in the category of non-sensitive attributes. In reality, an adversary can obtain information on quasi-identifiers of a target individual and exploits that knowledge to launch a privacy attack with the goal of inferring the value(s) of individual's sensitive attribute(s). One example of a relational data set is shown in Table 2.1. Attributes "Date of birth", "Gender", and "ZIP code" are quasi-identifiers and attribute "Disease" is sensitive.

- **Transaction data:** this type of data, also known as *market basket data*, has no fixed structure and it is often extremely high dimensional and sparse. It contains a collection of transactions and each transaction has its own set of items. One example of this type of data is the database of a grocery store containing the purchase information of customers. Each purchased product by a customer is an item and the set of all items bought by one customer is one transaction. A transaction data can be represented by a table in which each column is an attribute corresponding to one item and each row is one transaction. In the simplest form, for each transaction the values of those attributes that corresponds to the items of that transaction get value 1 and the other attributes get value 0. However, the

PID	Date of birth	Gender	ZIP Code	Disease
1	90/05/12	F	94142	Heart attack
2	90/10/13	F	94141	Hepatitis
3	90/05/15	F	94139	Heart attack
4	89/03/29	M	94139	Viral infection
5	89/07/18	M	94139	Cancer
6	90/01/3	F	94138	Hepatitis
7	92/09/27	F	94139	HIV
8	90/12/22	F	94141	Flu

Table 2.1: Relational data set

PID	Purchased items
1	milk, meat, banana, tea
2	bread, meat, tea, bear
3	milk, bread, banana
4	meat, eggs, bear, pregnancy test
5	bread, eggs, tea, adult toys
6	milk, bread, meat, banana, bear, tea

Table 2.2: Transaction data set

attributes can also be discrete or continuous. For instance, the number of each item purchased or the price of each item. Items can be either non-sensitive or sensitive. Non-sensitive items can act as quasi-identifiers to infer a sensitive item of an individual from his/her transaction. Any subset of items in a transaction is called an *itemset* and the number of transactions that contain an itemset i is called *support* of i. Table 2.2 shows a transaction dataset of a grocery store. "milk" and "eggs" are examples of non-sensitive items and "adult toys" and "pregnancy test" are sensitive items.

ID	Date	Time	Purchased items
1100	10/07/2013	8am	{Bread, Milk, Cereal}
1389	10/07/2013	9am	{Orange Juice, Egg, Bread}
1389	10/07/2013	7pm	{Yogurt, Milk}
1100	11/07/2013	9am	{Egg, Coffee}
1389	11/07/2013	2pm	{Banana, }

Table 2.3: Customer purchase history data

ID	Path
1	$\langle(a,1) \rightarrow (e,5) \rightarrow (f,4)\rangle$
2	$\langle(f,3) \rightarrow (a,7) \rightarrow (c,8)\rangle$
3	$\langle(b,3) \rightarrow (e,4) \rightarrow (d,6) \rightarrow (f,8)\rangle$
4	$\langle(b,2) \rightarrow (f,5) \rightarrow (a,6)\rangle$
5	$\langle(c,1) \rightarrow (d,6) \rightarrow (a,9)\rangle$
6	$\langle(d,4) \rightarrow (e,5) \rightarrow (g,9)\rangle$

Table 2.4: Trajectory data set

- **Sequence data:** this type of data contains sequences of elements or events related to individuals. There are two groups of sequence data: biological sequences (e.g. DNA, RNA, and protein) and event sequences (e.g. health data, clickstreams, customers purchase histories, etc.) [30]. In event sequence data a time stamp is often associated either with every record or with the events. In this type of data, in addition to non-sensitive attributes or items, the time stamps can act as quasi-identifiers. Table 2.3 shows an example of customer purchase history data in which each record consists of a customer identifier, date and time, and a set of items purchased. As it is pointed out in Chapter 1, longitudinal data is sequence data.

- **Trajectory data:** Trajectory data can be seen as a collection of sequences of spatio-temporal data points belonging to a moving object such as a GPS device, a cell phone, or an RFID tag. Any subset of times or locations or both acts as quasi-

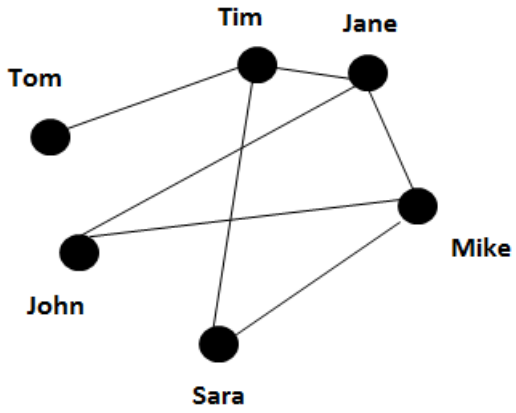


Figure 2.1: Graph data set

identifiers. One example of trajectory data is shown in Table 2.4. Each record in this dataset contains a path, which is a sequence of pairs (loc_i, t_i) specifying that an individual visited location loc_i at time t_i [81].

- **Graph data:** Graph data can be used to represent the relationships among instances. In this type of data, each node corresponds to one record and relationships between two records are represented by links. Based on the properties of the relationships, links can be directed/indirected and weighted/unweighted [114]. One example of a graph dataset is a social network in which each individual corresponds to one node and the friendship relationships among individuals are captured by links. Another example is a graph that represents the relationships among web pages through hyperlinks. In this type of data, any knowledge of adversaries about the relationships of individuals can act as quasi-identifiers. Figure 2.1 shows an example of a graph dataset representing the friendship relations among a number of individuals.

2.2 Anonymization Techniques

In an anonymization process, a dataset D is transformed to a new dataset D' through a set of modifications in order to prevent the disclosure of sensitive information. There are

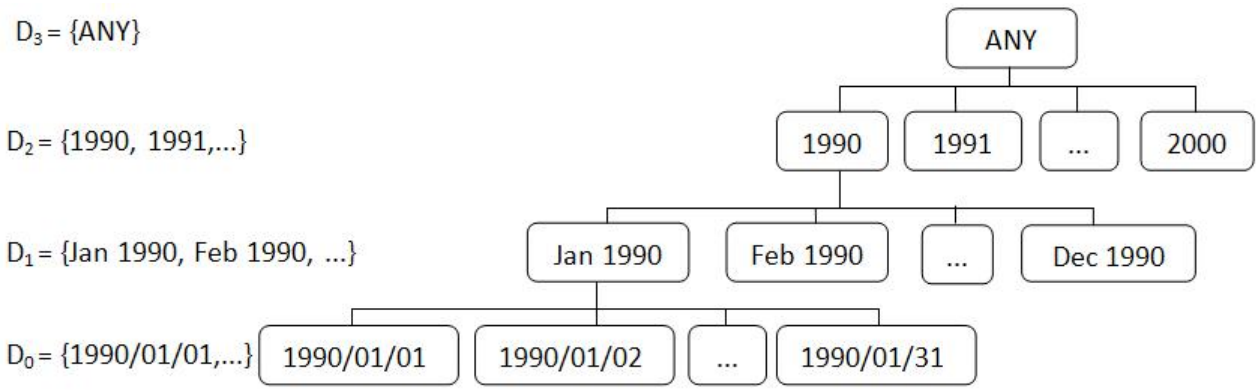


Figure 2.2: Generalization hierarchy for attribute "Date of birth"

several anonymization techniques, including *generalization*, *suppression*, *bucketization*, and *perturbation*. All these anonymization techniques produce a less precise dataset, however they are different in information loss they incur, the privacy protection they provide, and their search space [41].

2.2.1 Generalization

In generalization the specific values of those attributes that act as quasi-identifiers are replaced with more generalized ones. The goal is to increase the uncertainty of adversaries in linking a target individual to a record or sensitive information by hiding that individual in a group of records with the same quasi-identifier values. The group of all records with identical *QI* values is called an *equivalence class* (or *QI-group*) [127]. Generalization can be applied on both categorical and numerical attributes. For each categorical attribute usually there is a generalization hierarchy that represents the semantics of that attribute [127]. Using that hierarchy, a categorical value is replaced with a less specific categorical value that covers a larger domain of values for that attribute. For numerical attributes, values are replaced by a range of values. For instance, value 13 may be replaced by $[10, 20]$. Figure 2.2 shows a generalization hierarchy for attribute "date of birth" including all dates between year 1990 and year 2000.

The set of all possible generalizations for all attributes in *QI* form a generalization

lattice such that each node in this lattice corresponds to a possible generalization strategy. The optimal solution is the node that satisfies privacy requirements and results in the least information loss. To find the optimal solution, the naive approach is to enumerate all nodes of the generalization lattice and find the optimal node that gives the least amount of data distortion. However, it is shown in the literature that the problem of finding the optimal generalization solution is NP-hard [5, 80, 99, 126]. Therefore, heuristics are often employed to reduce the search space and find a sub-optimal solution. The goal is to find the *minimal generalization*, i.e., the minimum required set of modifications, that must be applied to a dataset to satisfy privacy requirements [110].

A value generalization can be applied to either *all* occurrences of one attribute value or just some instances. In the former, all instances of an attribute value are mapped to the same generalized value in the hierarchy of generalization. This is called *global recoding* [12, 57]. On the other hand, in the latter, different instances of the same attribute value may be generalized to different values in the generalization hierarchy. This approach is called *local recoding* [58, 49]. Local recoding gives more flexibility and has less data distortion compared to global recoding. However, since instances of an attribute value in the dataset generated by local recoding may have different generalized values, it may negatively affect the usefulness of the published data for analysis. In other words, it may cause inconsistencies in the data that may make it vulnerable to some privacy attacks such as *minimality attack* [18]. Moreover, the knowledge discovered from the resulting data may not be consistent with the original data and false results, which can mislead researchers, may be generated [41]. Global recoding, on the other hand, does not have this issue, since all occurrences of one value are generalized to the same generalized value.

At the attribute level, generalization can be applied in different ways, namely *full-domain generalization* [57, 99, 111], *Subtree generalization* [12, 42, 43, 50, 57, 18, 118], *sibling generalization* [57], and *cell generalization* [57, 126, 131, 49]:

- **Full-domain generalization:** in this scheme, all values of an attribute are generalized to the same level in the generalization hierarchy. In other words, all values of

one attribute are replaced with their generalized values from the same generalized domain. It has a smaller search space than the other approaches but it incurs the highest data distortion due to enforcing all attribute values to be generalized to the same generalized domain. For instance, in Figure 2.2, if values "1990/01/01" and "1990/01/31" are generalized to "Jan 1990", then all other values of date of birth in domain D_0 will also be generalized to their corresponding values in domain D_1, e.g. "1992/04/06" will be generalized to "Apr 1992". Full-domain generalization is a global recoding approach.

- **Subtree generalization:** Based on this scheme, when a value is replaced with the generalized value at node n, all other values in the leaves of the subtree rooted at n, will also be generalized to the value in node n. For example, in Figure 2.2, if "1990/01/01" is generalized to "Jan 1990" then all other values in the subtree rooted at "Jan 1990", i.e., "$1990/01/02, 1990/01/03, ..., 1990/01/31$", must be also generalized to "Jan 1990", but the values that are not in this subtree, such as "1992/04/06", do not need to be generalized. This approach is also a global recoding technique but has less information loss than full-domain generalization.

- **Sibling generalization:** This approach is similar to subtree generalization but it does not require that all sibling values in the leaves of a subtree to be generalized. For example, in Figure 2.2, some of the values in the subtree rooted at "Jan 1990" are generalized to "Jan 1990" but other values in that subtree remain ungeneralized. If we look at the resulting dataset from this scheme, for attribute "date of birth" we may see value "Jan 1990" and all values in the subtree of "Jan 1990" except "1990/01/02", "1990/01/13" and "1990/01/25" that means these three values are generalized to "Jan 1990". This is also a global recoding approach.

- **Cell generalization:** This approach is a local recoding technique in which generalization is only applied to some instances of some attribute values and the other instances remain intact. This technique incurs the least data distortion among all

generalization techniques but it has the same drawbacks as local recoding, pointed out above.

2.2.2 Suppression

Suppression changes the value of an attribute to a special value, such as "*" or "ANY", which means this value will not be released in the published data. In other words, that value is deleted from the dataset. Similar to generalization, suppression can be performed *locally* or *globally*. There are different schemes to apply suppression to a dataset, namely *record suppression* [12, 50, 57, 99], *value suppression* [120, 121, 18], and *cell suppression*(or *local suppression*) [80]. In a record suppression a record is entirely removed from a dataset. Value suppression refers to replacing *all* instances of an attribute value with "*". This is a global suppression. In cell suppression, just *some* instances of a value is suppressed in a dataset. So, it may lead to inconsistencies in analyzing data. Suppression can be seen as a special case of generalization in which a specific value is generalized to *ANY* [127].

2.2.3 Bucketization

The main idea of bucketization (also called *anatomization*) [129] is to partition records into buckets based on a grouping strategy and then de-associate quasi-identifiers from sensitive attributes by randomly permuting the sensitive values in each bucket. The exact values of quasi-identifiers and sensitive attributes are released in two separate tables. Both tables will have an attribute called *GroupID* and all records in the same bucket will have the same *GroupID* in both tables. As opposed to generalization and suppression, this technique does not modify the attribute values so the data recipient can use the original values for data analysis [127]. However, since it publishes the data in two tables, it will have some issues for applying standard data mining tools such as *classifiers* or *association rules* [41].

2.2.4 Perturbation

In perturbation, the attribute values are changed to synthetic values such that the statistical information that can be obtained from the generated dataset is very close to the original data. This approach has been mostly used in statistical disclosure control [123]. As opposed to generalization and suppression that preserve the truthfulness of the records, perturbation results in datasets in which the values of attributes are synthetic and, therefore, meaningless for data recipients. The resulting dataset can only be used for statistical analysis. The most common techniques for data perturbation are *value swapping* [26, 94, 95], *randomization* [6, 13, 31, 52], and *generating synthetic data* [3, 4].

- **Value swapping:** in this technique, two values of the same attribute in two records are exchanged. Value swapping does not change the domain of attributes, but the possible combinations of values of different attributes may not be the same as the original dataset. This may sometimes generate meaningless combinations. For example, the combination of values "male" and "waitress" for attributes "Gender" and "Occupation" does not make sense since "waitress" refers to a female [127].

- **Randomization:** This technique can be applied on numeric attributes. In this approach, data is perturbed with an appropriate level of noise, randomly chosen from a distribution, which can be either added to or multiplied by the original value of each attribute [116]. The advantage of this technique is that it preserves some statistical properties such as mean and correlation [54], but it may generate some meaningless values [127]. Moreover, additive noise may be filtered by the adversary and she will be able to closely estimate the real values of numeric sensitive attributes [52]. Multiplicative noise does not have this drawback.

- **Generating synthetic data:** in this technique, first a statistical model is generated from dataset and then a synthetic data is built which follows that model. The advantage of this technique is that all statistical properties of the original data are

preserved. However, it may generate some meaningless values that does not exist in the real world [127].

2.3 Information Loss Metrics

Data anonymization incurs some information loss and destroys data utility. In order to preserve the usefulness of published data, it must be ensured that the minimum data distortion occurs in the process of data anonymization. There are several metrics to measure the amount of information loss. These metrics can either be used to measure the utility of published data with respect to original data or be used as a search metric to guide the steps of searching for the best anonymization solution in the search space of all possible solutions [41]. The selection of an appropriate information loss metric mostly depends on the anonymization algorithm as well as the goal of data publishing. For instance, if an anonymization algorithm uses a generalization hierarchy, then the information loss metrics which take into account the cost of generalization and suppression operations are of interest; if anonymized data is being published for a specific data mining task, such as classification, then the information loss metrics which consider such specific goals are able to better capture the incurred information loss.

Some of these metrics are suitable for general purpose data publishing scenarios, while some other measures are defined to capture information loss when data is released for a specific task, for instance, *building a classifier*. Also there are a few metrics that, in addition to information, take into account the privacy requirements to measure the information loss of anonymization.

The metrics in the first category, i.e., general purpose metrics, measure the usefulness of data in the scenarios that there is no pre-defined specific goals for using published data. Therefore, the data publisher must anonymize data so that the maximum data utility to be preserved to make data applicable for *any* type of analysis. Intuitively, the "similarity" of published data to the original data can be a good measure to quantify

the information loss in this scenario.

Every generalization or suppression cause some data distortion. The *Distortion* measure (Distortion) [99, 108, 110] captures data distortion by assigning a penalty to every instance of an attribute value which is generalized or suppressed. The idea is that if the value of an attribute in a record, i.e., a cell in the dataset, is not generalized or suppressed there is no distortion. In every generalization to a value at a higher level in the hierarchy, distortion will increase. Therefore, the distortion for each cell is defined as the height of the node corresponding to the generalized value in the generalization hierarchy divided by the maximum height of the hierarchy. For example, in Figure 2.2, if value "1990/01/03" in one record is generalized to "Jan 1990", the distortion of that cell due to this generalization is $\frac{1}{3}$. The total distortion of the whole dataset is the sum of all cell distortions. Also, one minus the sum of all cell distortions (normalized by the total number of cells) is defined as its *Precision*(*Prec*).

Another information loss measure is the *Discernability Measure*(*DM*) [12, 74, 36] that penalizes each record for being indistinguishable from other records. For each record r in an equivalence class, the information loss is the size of the equivalence class. However, since this measure is based on the size of equivalence classes, it gives the same information loss for all records in the equal-sized equivalence classes. But those records may be generalized differently and, therefore, have different levels of distortion which can not be captured by *DM* [127]. Another drawback of *DM* is that it can not effectively capture the information loss when the attributes in a dataset are non-uniformly distributed [37, 62].

There are a number of works that consider the number of leaf nodes in the subtree rooted at each generalized value in the hierarchy of generalization to measure the amount of information loss due to generalization and suppression. One of the first works that proposed such information loss measure was [50]. In this work, Iyengar introduced the *General Loss Metric*(*LM*) that captures the information loss by summing up the normalized information loss for each attribute. Let M_P denotes the number of leaf nodes

in the subtree rooted at node P and M denotes the total number of leaf nodes in the generalization hierarchy for categorical attribute A. Then the normalized information loss for each value of A which is generalized to P is defined as $\frac{M_P-1}{M-1}$. If A is a numerical attribute, then the normalized information loss for every value of A which is generalized to an interval $[L_i, U_i]$ is defined as $\frac{U_i-L_i}{U-L}$ where U and L are the maximum and the minimum values for A in the dataset, respectively. The normalized information loss for each attribute is computed by averaging the loss for each of its values. For example, using the generalization hierarchy in figure 2.2, the normalized information loss of generalizing one instance of value "1990/01/03" of attribute "Date of birth" to "Jan 1990" is approximately $\frac{31-1}{3650-1} = 0.0082$ (in a 10 years space when leap years are not considered).

A similar information loss measure to LM is the *Normalized Certainty Penalty* (NCP) [131] which defines the information loss as:

$$NCP(T) = \sum_{t \in T} \sum_{i=1}^{n} (w_i \cdot NCP_{A_i}(t)) \tag{2.1}$$

where n is the number of QI attributes in dataset T, w_i denotes the importance of attribute A_i and $NCP_{A_i}(t)$ is the normalized certainty penalty of the value of attribute A_i in record t, defined as

$$NCP_A(t) = \frac{size(u)}{|A|} \tag{2.2}$$

where $size(u)$ is the number of leaf nodes of the subtree rooted at u and $|A|$ is the number of distinct values of attribute A.

Another measure similar to LM and NCP was presented in [128, 130]. Let v^* be a node in the generalization hierarchy H of an attribute $A \in QI$. The *coverage* of v^* in a generalization hierarchy, denoted by $coverage[v^*]$, is defined as the number of leaves in the subtree rooted at v^* and the *base* of H, denoted by $base(H)$, is defined as the total number of leaves of H. The information loss of generalizing a specific value to v^* is defined as:

$$ILoss(v^*) = \frac{coverage[v^*] - 1}{base(H)} \tag{2.3}$$

Let t^* be a generalized record and v_A^* is the generalized value for attribute A in t^*. The information loss of a generalized record t^*, is defined as:

$$ILoss(t^*) = \sum_{A \in QI} \{ILoss(v_A^*) \times weight(A)\} \tag{2.4}$$

where $ILoss(v_A^*)$ is measured based on Equation 2.3 and *weight(A)* is a weight parameter that can be specified by a user to determine the importance of each attribute. The total information loss in the anonymized dataset T^* will be:

$$ILoss(T^*) = \frac{\sum_{t^* \in T^*} ILoss(t^*)}{|T^*|} \tag{2.5}$$

where $|T^*|$ is the number of records in T^*.

All of the metrics, discussed above, are general purpose metrics. However, as we pointed out at the beginning of this section, there are some metrics which are introduced to measure the information loss when a dataset is published for a specific purpose. These measures take into account the purpose of data publishing and aim to retain the information which is essential for that specific task. For example, if the data is released for building a classifier for a target attribute, the values that are important for classification should not be generalized or suppressed. To achieve this goal, the classification error on the future instances must be considered in measuring the information loss [41]. However, since future instances are unknown at the time of data anonymization, the training cases in the dataset are used for calculating the accuracy of classifier [42, 43, 50]. The most commonly used measure in this category is *Classification Measure* (CM) [50]. This metric penalizes each record r that is suppressed or generalized to an equivalence class in which the class label of the record is not the majority class:

$$CM = \sum_{r \in D} \frac{Penalty(r)}{N} \tag{2.6}$$

where D is the dataset, N is the number of records in D and *Penalty(r)* is defined as follows:

$$Penalty(r) = \begin{cases} 1 & \text{if } r \text{ is suppressed} \\ 1 & \text{if } class(r) \neq Majority(EC(r)) \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

where *Majority(EC(r))* indicates the majority class in the equivalence class of r.

As we pointed out, there are some metrics that consider both information and privacy. The goal is to choose the anonymization that minimizes the information loss while maximizing the privacy gain. One of such measures is defined in [42, 43] as:

$$ILPG(g) = \frac{IL(g)}{PG(g) + 1} \tag{2.8}$$

where IL(g) is the information loss and PG(g) is the privacy gain by applying generalization g. The goal is to select the generalization that minimizes *ILPG(g)*.

Chapter 3

Privacy Preserving Data Publishing: A Literature Review

In this chapter, we will review state of the art in privacy preserving data publishing. As we discussed in Chapter 2, there are three different types of disclosure, including *identity disclosure*, *attribute disclosure*, and *membership disclosure* and we study privacy preserving approaches for publishing data with respect to these types of disclosure. However, as one of the primary assumptions in our data publishing scenario is that *the adversary knows that the record of the potential victim is in the released dataset*, membership disclosure will not happen in our scenario. Therefore, in this chapter we only consider identity disclosure and attribute disclosure. In each of these categories, at first, we talk about the types of background knowledge an adversary may pose as well as the potential privacy attacks using such knowledge. Then, we discuss the proposed solutions in the literature to prevent potential privacy attacks. We classify these methods into three categories based on the structure of the data being de-identified, including relational data, transaction data, and trajectory and sequence data.

3.1 Identity Disclosure

Identity disclosure, also known as *record linkage* [41], happens when an adversary links an individual to a small number of records which match his background knowledge and consequently infers some sensitive information about that individual. In the following, we first study the type of background knowledge an adversary can employ to re-identify an individual's record as well as the potential attacks based on that knowledge. Then, we describe the proposed methods aiming to protect data against identity disclosure.

3.1.1 Background Knowledge of the Adversary

In an identity disclosure attack, the goal of the adversary can be either to re-identify the record of a specific individual or an *arbitrary* individual whose information is in the released data set. More precisely, in the former, the adversary knows that the record of a specific individual is in the released data and she employs her background knowledge about that individual to re-identify the individual's record. In the latter, without targeting any specific individual, the adversary attacks data with the goal of re-identifying *any* individual.

To prevent identity disclosure, a common practice has been to remove the explicit identifiers such as *names*, *social security numbers*, *phone numbers*, and *addresses* from data before release [111]. However, Sweeney [111] showed that simply removing explicit identifiers is not sufficient to protect the privacy of individuals against identity disclosure. Based on the results of her work in [109], she showed that 87% (216 million of 248 million) of the population in the United States can be uniquely or nearly uniquely identified by combination of only their *date of birth*, *gender* and *5-digit ZIP code*.

Attributes like date of birth, race, ZIP code and gender are called *quasi-identifiers* (QI). A quasi-identifier is a set of attributes that can be linked with external information to uniquely (or nearly uniquely) re-identify the record of an individual in the released data [22]. Removing only explicit identifiers from a data set may not adequately protect

data as long as some quasi identifiers exist in the records of individuals in a released data set [111].

In addition to relational data, where there is an explicit set of quasi identifiers, other types of data such as transaction data, sequence data, and trajectory data are vulnerable to identity disclosure. In transaction data, every subset of items can potentially act as a quasi identifier to uniquely or nearly uniquely re-identify an individual's transaction. A sequence database is susceptible to an identity disclosure if an adversary knows a specific sub-sequence of events or elements of an individual whose record is in the released sequence data. In a trajectory data, a sub-sequence of spatio-temporal data points belonging to a moving object, such as a GPS device in an individual's car, may be known by an adversary and be employed by her to re-identify the record of that individual.

In all these cases, the adversary can get such knowledge from different external sources. She may know the victim, for instance, she can be the victim's neighbor and, therefore, she will know some information about the victim such as his/her age, ZIP code, and gender. She may also have access to a public external database containing the record of the victim. She may even find some information accidentally; for instance, she may see her neighbor on the bus and find out some of the items that he bought from the supermarket.

As we pointed out earlier in this section, the adversary may launch a privacy attack on a released data set with the goal of re-identifying the record of a specific or arbitrary individual. She can leverage her background knowledge to increase the success of her attack. In the next sections, we will talk about privacy attacks leading to identity disclosure and methods to protect data against such attacks.

3.1.2 Privacy Attacks

Having some background knowledge about an individual, an adversary may be able to uniquely, or nearly uniquely, re-identify a victim's record in the released data. For this purpose, the adversary uses her background knowledge to *link* that individual to a record

in a (supposedly) de-identified dataset. This kind of attack is called a *linking attack* [20] and was described by Sweeney [111] through the following example.

Example (A linking attack on relational data) [111]. *The Group Insurance Commission(GIC) in Massachusetts which is responsible for purchasing health insurance for all Massachusetts's employees, after removing all explicit identifiers, released a copy of its data set containing the medical information of all employees in Massachusetts as well as their demographic information such as ZIP code, birth date, and gender. Beside having access to this data set, Sweeney purchased a copy of voter registration list for Massachusetts which is a public data set containing name, address, ZIP code, birth date, and gender of each voter. By linking these two data sets on the shared attributes ZIP code, birth date, and gender, Sweeney identified the medical record of former governor of Massachusetts, William Weld, including all diagnoses, procedures and medications.*

In addition to relational data, the other types of data, like transaction data, sequence data, trajectory data, etc., are also vulnerable to linking attacks. In the following, through a set of examples, we illustrate linking attacks on different types of data.

Example (A linking attack on transaction data). *Assume that a hospital wants to publish its data set containing transactions of patients who visited the hospital during last year. Each transaction consists of a set of diagnosis corresponding to each patient. The hospital removes all identifying information like names and social security numbers and releases the anonymized data set. An example of this data is shown in Table 3.1. Assume that Alice knows that her neighbor, Bob, has been in this hospital 2 months ago because of disease "d". If Alice gets access to the released data of the hospital, she will be able to simply infer that record 4 belongs to Bob since this is the only record in this database that contains disease "d".*

In the case of sequence data, if the adversary knows a sub-sequence of actions about an individual, she may be able to re-identify that individual's record and, consequently, infer the individual's whole sequence. Pensa et al [89] showed this issue with the following example:

PID	Diagnosis Codes
1	a, c, f, h
2	b, c, h, j, k
3	a, b, f, k
4	c, d, g, i, j
5	b, g, h, i
6	a, b, c, f, g, h

Table 3.1: Patient data

Example (A linking attack on sequence data) [89]. *Assume that Alice has access to the city traffic data. If she knows that Bob often goes from the commercial zone A to the general hospital B and the sequence $A \Rightarrow B$ occurs few times in the data set, she will be able to easily identify Bob's sequence record in the released city traffic data. This will enable her to infer the entire sequence of places that Bob visited during the day and, therefore, breach his privacy.*

Publication of a trajectory data may also threat the privacy of individuals if the adversary knows a specific path of spatio-temporal data points related to an individual. To illustrate this, we consider the example from [81]:

Example (A linking attack on trajectory data) [81]. *Assume that a hospital needs to release the patient-specific trajectory and health data shown in Table 3.2. Each record in this table contains a path, which is a sequence of pairs (loc_i, t_i) specifying that the patient visited location loc_i at time t_i. In addition, each record contains the sensitive attribute Diagnosis. A linking attack happens on this data set, if there is a specific path which is not shared with many patients. Assume that Alice knows that Bob visited locations e and g at times 5 and 9, respectively. Since there is only one record, ID = 6, that contains (e,5) and (g,9), Alice will uniquely re-identify Bob's record. Consequently, she will find out the other locations Bob visited as well as his disease, i.e., HIV.*

In all the above examples, a linking attack occurred due to an exact match between the piece of background knowledge of the adversary about an individual and one part of

ID	Path	Diagnosis
1	$\langle(a,1) \rightarrow (e,5) \rightarrow (f,4)\rangle$	Flu
2	$\langle(f,3) \rightarrow (a,7) \rightarrow (c,8)\rangle$	Diabetes
3	$\langle(b,3) \rightarrow (e,4) \rightarrow (d,6) \rightarrow (f,8)\rangle$	Fever
4	$\langle(b,2) \rightarrow (e,5) \rightarrow (g,6)\rangle$	viral infection
5	$\langle(c,1) \rightarrow (d,6) \rightarrow (a,9)\rangle$	Flu
6	$\langle(d,4) \rightarrow (e,5) \rightarrow (g,9)\rangle$	HIV

Table 3.2: trajectory and health data set

that individual's record in the released data. However, a linking attack may also happen based on the *semantic* information contained in the record of an individual, for instance, in his/her search queries [20]. To illustrate this kind of linking attack, we consider the case of AOL.

Example (A linking attack based on semantic information) [11, 20]. *On August 6th, 2006, AOL released a data set containing 20 million web queries from 650,000 of its users collected over a period of three months. The data set also contained the URL clicked from each search result and its ranking. To anonymize the data before release, AOL replaced each user ID with a randomly generated number. Two New York Times reporters analyzed this data set based on the semantic content of users' search queries such as the name of a town, age-related queries, several searches with a particular last name, etc, and, after three days, they could track down user number 4417749, Thelma Arnold, a 62 years old woman living in Lilburn, GA, and interview with her. In her search query logs, Thelma had hundreds of searches over a period of three months on topics like "60 single men", "dog that urinates on everything", and "numb fingers".*

In addition to these examples, linking attacks can lead to a privacy breach in the other domains including social networks [10, 48, 83] and genomic records [75, 76, 77]. In our literature review, we only focus on privacy attacks on relational data, transaction data, and trajectory and sequence data. In the next section we will talk about the proposed approaches in the literature to prevent identity disclosure through linking attacks.

3.1.3 Privacy Models to Prevent Identity Disclosure

As we discussed in Sections 3.1.1 and 3.1.2, identity disclosure occurs when the adversary can re-identify the record of an individual in a released dataset either with or without employing her background knowledge. In this section, we review the privacy models proposed in the literature to protect data against identity disclosure. We classify these methods into three categories based on the structure of the data which will be released. i.e., relational data, transaction data, and trajectory and sequence data.

3.1.3.1 Relational Data

An identity disclosure in relational data can happen when an adversary finds a (or a few) match(es) in the released data set for those quasi identifiers she knows about an individual. To avoid this type of disclosure, several techniques are proposed in the literature [23] among which sampling, swapping values and randomization have been some of the most common approaches. However, in all these techniques data is modified such that the correctness of individual records is compromised. Therefore, these techniques are inappropriate in the applications where the "truthfulness" of the released data is required [22]. An alternative approach to deal with this limitation is data anonymization. In this technique, individuals' identifying information is either removed or altered to ensure the anonymity of individuals. The most common approach in data anonymization is the notion of *k-anonymity* [100, 101, 111] which was proposed by Samarati and Sweeney.

***k*-anonymity.** This privacy model not only protects data against identity disclosure but also preserves the truthfulness of the data. A data set satisfies *k-anonymity* iff for every combination of values of quasi identifiers, there are at least k records in the data set sharing those values. In other words, each record in a *k-anonymous* data set is indistinguishable from at least k-1 other records with respect to a set of quasi identifiers [100, 101, 111, 22]. In a k-anonymous dataset the probability of linking an individual to a specific record with respect to the values of quasi identifiers is at most $\frac{1}{k}$ [41]. Table 3.3 shows an example of a 3-anonymous data set where ZIP code and date of birth are

PID	Zip Code	Date of Birth	Disease
1	120**	1975	Heart Disease
2	120**	1975	Bronchitis
3	120**	1975	Viral Infection
4	120**	1970	Viral Infection
8	120**	1970	Cancer
9	120**	1970	Cancer
5	118**	1973	Cancer
6	118**	1973	Heart Disease
7	118**	1973	flu
10	118**	1973	Diabetes

Table 3.3: 3-anonymous data

quasi identifiers.

In order to make data k-anonymous, the first step is to recognize the set of quasi identifiers in the data set. Choosing quasi identifiers depends on determining what external sources of information an adversary may have to launch a linking attack [22]. The simplest assumption, which was made in the original k-anonymity paper [100] and most of its refined versions, is to consider a single quasi identifier consisting of all attributes that can potentially exist in the external sources and can be employed by an adversary for linking attacks.

Although this assumption provides more protection due to considering more attributes as quasi identifiers, this will lead to more data distortion since the records in an equivalence class must agree on more attributes to satisfy k-anonymity [41]. Fung et al [42, 43] addressed this problem by considering multiple sets of quasi-identifiers. According to their work, the data must be k-anonymous with respect to every set of quasi identifiers. For instance, if there are two sets of quasi identifiers QI_1 and QI_2, then each record must be indistinguishable from k - 1 other records with respect to both QI_1 and QI_2. Those k - 1 other records can be different for each set of quasi identifiers. The only challenge in applying this technique is that the data publisher needs to know how and

based on what information the adversary will do a linking attack. Otherwise, this may cause higher data distortion or more disclosure risks [41].

To achieve k-anonymity of a data set, the original model [100] and most of its improved subsequent versions [12, 50, 57, 110, 125, 37] employed *generalization* and *suppression*. These two anonymization techniques, unlike the other approaches like swapping and adding noise, retain the truthfulness of the records in the data, and, therefore, satisfy the main goal of k-anonymity [22].

In most versions of k-anonymity it is assumed that a single table needs to be anonymized. However, there are some cases when a database contains multiple relational tables and, therefore, those methods either fail to anonymize that database or incur a high information loss to make the data k-anonymous [86]. To deal with this limitation Nergiz et al proposed the notion of *MultiR k-anonymity* [86] .

MultiR k-anonymity. In this model, the authors assume a database containing a person specific table PT and a set of tables $T_1, T_2, ..., T_n$. PT has an identifier attribute *pid* and some sensitive attributes and each table $T_i, 1 \leq i \leq n$, contains a set of quasi-dentifiers and sensitive attributes as well as some foreign keys. According to this privacy model, the data is k-anonymous if for each individual o corresponding to the join of all tables $PT \bowtie T_1 \bowtie ... \bowtie T_n$, there are at least k - 1 other individuals who have the same values of quasi identifiers as o [41].

All methods we discussed so far are proposed to prevent identity disclosure in a relational dataset. In the next section, we discuss some methods focusing on transaction data.

3.1.3.2 Transaction Data

As we discussed in Section 3.1.2, in the case of transaction data every subset of items can potentially act as a quasi-identifier and an adversary, who knows a subset of related items to an individual, can launch a linking attack against that individual in a released transaction data.

The challenge in anonymizing transaction data is that it does not have any fixed structure and it is often extremely high dimensional and sparse. For instance [41], Amazon.com has millions of catalog items and, therefore, a dataset containing purchase records of the Amazon's customers will be very high dimensional. In order to anonymize this dataset, every possible combination of these several million items must be considered as a potential quasi-identifier that can be used by an adversary for privacy attacks. In the presence of a dimensionality problem, as shown in [2], most of the data must be modified (by suppression, generalization,...) in order to satisfy privacy requirements. But this will incur extremely high information loss and will make the data useless. To deal with this challenge some specific techniques are proposed to protect transaction data against identity disclosure and attribute disclosure. In this section we review the proposed techniques to prevent identity disclosure. The methods for attribute disclosure will be discussed in Section 3.2.3.2.

K^m-anonymity. Terrovitis et al proposed the notion of *k^m-anonymity* [118] to k-anonymize transaction data. In order to deal with data dimensionality, the authors assumed that the knowledge of the adversaries about transactions is limited. This is a rational assumption since in practice it is not feasible for an adversary to know all items belonging to a victim and it is more likely that she has partial background knowledge about the transactions. Assuming that the knowledge of the adversary is at most m items in a specific transaction, Terrovitis et al defined k^m-anonymity model which ensures that for any possible itemset of size at most m, there are at least k transactions that contain that itemset. To enforce *k^m-anonymity*, Terrovitis et al employed a *global generalization* technique.

Liu et al [65] showed that using global generalization to make the data k^m-anonymous incurs extreme information loss in the presence of outliers. Two more drawbacks of k^m-anonymity were brought up by He et al [49]. First, they argue that there may be some cases when the data publisher will not be able to specify the maximum knowledge of the adversary, so he can not determine a safe value for m. Second, they illustrated that

k^m-anonymity may fail to protect against some attacks based on general background knowledge. To address these challenges, He et al [49] proposed a top-down approach based on *local generalization* to anonymize transaction data.

Transactional *k*-anonymity. This privacy model [49], which is similar to k-anonymity model on relational data, ensures that for every possible itemset, there are at least k transactions containing that itemset. In other words, a transaction dataset is k-anonymous if each transaction in this dataset is identical to at least k - 1 other transactions [49].

If a transaction data set satisfies k-anonymity, then it satisfies k^m-anonymity for any value of m. However, as He et al argue, even for m = m_{max}, where m_{max} is the maximum length of a transaction, k^m-anonymity can not provide the same level of protection as k-anonymity [49]. Also, unlike k^m-anonymity model that fails against attacks based on public knowledge, transactional k-anonymity model protects the data against such attacks since it guarantees that every transaction is indistinguishable from at least k - 1 other transactions. He et al proposed a top-down partition-based algorithm, using local generalization, to enforce transactional k-anonymity.

Later Liu et al [65] argued that transactional k-anonymity results in excessive distortion to the data. This is due to the fact that in transactional k-anonymity, data is anonymized with respect to every possible itemset. But, in practice, most of these itemsets may be beyond the knowledge of the adversary. So, they will not be a threat to the privacy of individuals and do not need to be considered in data anonymization. Liu et al also showed that the results of analyzing the anonymized data based on local generalization technique, employed in transactional k-anonymity, will not be easily interpretable. To deal with these issues, Liu et al introduced an anonymization approach that enforces k^m-anonymity to transaction data by integrating the *global generalization* technique with the *global item suppression* technique. This approach can also deal with the problem of k^m-anonymity with respect to global generalization, since the outliers can be removed by suppression.

Another work in this category was proposed by Loukides et al [68, 69]. The authors argued that considering all item sets up to a certain length as potential privacy threats, like the approaches in [118] and [65], is not effective in the cases when only some specific itemsets, consisting of certain items and varying in size, are threatening the privacy of individuals in a data publication scenario. In those cases, approaches like [118] and [65] anonymize the data unnecessarily and incur a significant data distortion. Moreover, in a real life application, it may be required that some specific items, that, for instance, are of interest of a research work, be preserved in data anonymization [68, 69]. Loukides et al showed that none of the prior approaches for anonymizing transaction data can satisfy such specific utility requirements. To address these issues, Loukides et al proposed a *constraint-based anonymization technique* (*COAT*). They consider a set of privacy constraints and utility constraints based on the specific requirements of the problem and anonymize data such that each transaction will be indistinguishable from at least k-1 other transactions with respect to privacy and utility constraints. For data anonymization, they use both generalization and suppression. However, instead of a hierarchy-based generalization, they introduced a set-based generalization technique in which each item i is mapped to a non-empty subset of items that contains i. That non-empty subset will be considered as a generalization of i.

As this work is inspired by k-anonymity, it has the same limitation of k-anonymity in dealing with attribute disclosure. Moreover, the resulting dataset from a set-based generalization and its analysis results may not be easily readable and interpretable. Since each transaction in the released data may contain several subsets of items which are generalizations of the original items in those transactions.

In [46], authors proposed a clustering-based anonymization framework to anonymize transaction data with low information loss. The proposed framework is independent of the data transformation method and can anonymize transaction data under various privacy and utility constraints similar to ones defined in [68, 69]. Authors presented two anonymization algorithms which employ generalization and a combination of generaliza-

tion and suppression-based heuristics, respectively.

3.1.3.3 Trajectory and Sequence Data

Trajectories of mobile objects such as individuals and cars can be easily collected through technologies like GPS and $RFID$. An adversary who knows a sub-sequence of locations, times or both in spatio-temporal data points visited by a target individual may violate individual's privacy by linking her knowledge to published trajectories and re-identify trajectory of target individual. Publication of sequence data may also violate privacy of individuals. In a published sequence data every subsequence of events belonging to a target individual may be known by an adversary and can lead her to re-identify the full sequence of that individual. Several anonymization techniques have been proposed to protect trajectory and sequence data against identity disclosure.

(k, δ)-anonymity. [1] The general idea in this method is to modify trajectories such that in every cylinder of the radius δ there are at least k trajectories. For this purpose, trajectories are first divided into disjoint groups such that all trajectories in a group have approximately the same start time and end time. Then trajectories in each group are clustered using the Euclidean distance. Every cluster will contain at least k trajectories with Euclidean distance δ. One limitation of this method is the assumption of continuous trajectories which does not hold for all moving objects, such as RFID data. Also, the first step of the algorithm which partitions trajectories into disjoint groups may result in small groups with less than k trajectories. Moreover, the data is anonymized by changing the actual location of objects which means the truthfulness of data is not preserved .

Another method to k-anonymize trajectory data was proposed by Nergiz et al [85]. This method partitions trajectories into groups of size at least k by using a clustering approach which minimizes a log cost metric. Data is anonymized by exploiting space and time generalizations, point matching in space and time, and suppressing points and trajectories.

In [89], a k-anonymity model for anonymizing sequence data is proposed. In this

method, sequences are modified by insertion, deletion or substitution of items to generate groups of k indistinguishable sequences. Data is anonymized with the goal of preserving frequent sequential patterns. This work destroys truthfulness of data due to item insertion and substitution.

Terrovitis et al. [117] assumed that different adversaries may possess different background knowledge and that the data publisher is aware of all such adversarial knowledge. This method aims to modify data such that none of these adversaries can link their knowledge to fewer than k individuals. The authors proposed a suppression-based approach which removes minimum number of items in order to make data k-anonymous. However, the assumption of knowing all potential background knowledge of adversaries before data publication, is not feasible in many trajectory data publishing scenarios.

Yarovoy et al. [134] considered timestamps as quasi-identifiers and assumed that each trajectory has its own set of times as its quasi-identifier. Based on this assumption, they proposed a notion of k-anonymity for trajectory data by defining an attack graph for adversaries. A moving object satisfies k-anonymity if the attack graph is symmetric and every node in the graph has degree k. k-anonymity is achieved by identifying equivalence classes and generalizing all records in every equivalence class to common regions with respect to quasi-identifiers. The authors assumed that the data publisher knows the quasi-identifiers for each trajectory, but it it not mentioned how this knowledge can be obtained.

Poulis et al [91] proposed an anonymization method for trajectory data based on k^m-anonymity model and using distance-based generalization. They argued that a location taxonomy may not adequately reflect the distance between locations and generalizing trajectories based on these taxonomies may lead to high information loss. Instead, they proposed using generalized locations defined as sets of at least two locations. If an anonymized trajectory t contains a generalized location $L = \{l_1, \ldots, l_v\}$, it means that it contains exactly one location in $l_1, \ldots, l_v$ in the original data. The proposed anonymization algorithm works in an apriori-like fashion based on the apriori principle [115].

All methods discussed in this section and Sections 3.1.3.1 and 3.1.3.2 protect data against identity disclosure by ensuring that each individual is indistinguishable from a large group of other individuals in the released data set. However, in some cases, an adversary is able to infer some sensitive information about a target individual without re-identifying that individual's record, for instance when most of the records in an equivalence class have the same sensitive value. This kind of disclosure is called attribute disclosure. In the next section, we will discuss the proposed techniques to prevent attribute disclosure.

3.2 Attribute Disclosure

Attribute disclosure occurs when the adversary can infer some sensitive information about an individual without identifying that individual's record in the published dataset. To address this issue several privacy models were proposed with the goal of increasing the uncertainty of the adversary in deriving sensitive information from published data. In this section we review the existing methods in the literature addressing the problem of attribute disclosure. As done in Section 3.1, first we describe different forms of background knowledge of the adversary in an attribute disclosure scenario. Then we talk about the potential privacy attacks that occur based on each type of background knowledge of the adversary. Then we review the approaches introduced in the literature to tackle attribute disclosure attacks. We classify these techniques into three categories based on the type of data these techniques are applied to: relational data, transaction data, and trajectory data.

3.2.1 Background Knowledge of the Adversary

The simplest form of background knowledge an adversary may pose is the values of quasi-identifiers of an individual. Employing this knowledge, the adversary may be able to directly infer an individual's sensitive value(s) based on the distribution of values

of sensitive attribute(s) in the equivalence class where individual's record belongs to. Besides quasi-identifiers, an adversary may have more complex forms of background knowledge that enables her for attribute disclosure attacks.

Machanavajjhala et al. [74] considered the background knowledge that can be modeled by *negation statements* [79]. For instance, "men do not have cervical cancer", or "Bob never travels, thus he is extremely unlikely to have Ebola", or "Japanese have a very low incidence of heart disease" [74].

Another form of background knowledge of the adversary leading to attribute disclosure is proposed by Li et al [63]. The authors mine negative association rules from the data as the background knowledge of the adversary and then anonymize the data in such a way that eliminates this knowledge from the data. The idea is that the background knowledge of the adversary reflects itself in the data; therefore, data mining approaches should be able to extract this knowledge. Although in this technique only the negative association rules are considered as the background knowledge of the adversary, Li et al talk about the possibility of discovering the other types of background knowledge from the data as long as it does not have any negative effect on the utility of data [63].

Wong et al [128] considered adversaries with the knowledge of the mechanism or anonymization algorithm employed for publishing data. They recognized that based on such knowledge and the fact that the main goal of all anonymization techniques is to reduce information loss, the adversary may be able to infer sensitive information.

Having *probabilistic knowledge* about one part of the domain can also empower the adversary for attribute disclosure attacks [20]. For example, the adversary may know the distribution of values of a sensitive attribute in a (or part of a) population, such as "the rate of cancer in Gotham City is only 10%, but it is higher (about 50%) if only males in Gotham City are considered" [20].

In the next section, we talk about the potential privacy attacks based on different background knowledge of the adversaries presented in this section.

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

Table 3.4: Original data [74] ©2006 IEEE

3.2.2 Privacy Attacks

Different types of privacy attacks, leading to disclosure of sensitive value(s) of individuals, are introduced in the literature. These attacks will be discussed in this section.

Homogeneity attack [74]. When all or majority of the records in an equivalence class in an anonymized data set have an identical value for a sensitive attribute the data set is vulnerable to a homogeneity attack. In this attack, the adversary uses her background knowledge about the quasi-identifiers of an individual to find the equivalence class where the individual's record belongs. Then, if all or most of the sensitive values in this equivalence class are the same, e.g. s_m, without needing to re-identify the record of that individual the adversary will be able to infer that the value of that sensitive attribute for that individual is s_m. This attack is illustrated by Machanavajjhala et al. [74] through the following example:

Example (homogeneity attack)*: Alice and Bob are neighbors and one day Alice sees that Bob is taken to the hospital by ambulance. While Bob is in the hospital, Alice discovers Table 3.5, a 4-anonymous version of the dataset in Table 3.4, containing current inpatient records published by the hospital. Therefore, she knows that one of the*

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Table 3.5: 4-anonymous data [74] ©2006 IEEE

records in the table belongs to Bob and she tries to figure out what Bob's disease is. As she is Bob's neighbor, she knows that Bob is American and he is 31 years old. She also knows that he is living in the zip code 13053 (the same as herself). Therefore, she can infer that one of the records in the last equivalence class, i.e., records 9, 10, 11, or 12, belongs to Bob. As all patients in that equivalence class have cancer, Alice, without any extra effort to re-identify Bob's record, will infer that Bob has cancer and therefore she will jeopardize his privacy.

The above example shows that when there is lack of diversity in the values of a sensitive attribute in an equivalence class the privacy of the individuals can be violated.

Background knowledge attack [74]. Adversaries can launch this attack if, besides quasi-identifiers, they pose some extra background knowledge about an individual. Machanavajjhala et al. [74] consider adversaries whose knowledge is in the form of negation statements. With such knowledge, an adversary is able to eliminate some (in special case, all except one) of the sensitive values in an equivalence class and therefore increase her certainty about the value of sensitive attribute of an individual belonging to that equivalence class. The authors demonstrated this attack with the following example in

which the adversary was able to eliminate all sensitive values in an equivalence class except one value, using her background knowledge:

Example (background knowledge attack) [74]*: Alice has a pen-friend named Umeko who is a 21 years old Japanese girl living in the zip code 13068. Alice knows that Umeko is admitted to the same hospital as Bob and, therefore, her record is in the published 4-anonymous data in Table 3.5. By looking at the data, Alice will know that Umeko's record resides in the first equivalence class and one of the records 1, 2, 3, or 4 belongs to Umeko. Without any extra information, Alice can not conclude if Umeko has viral infection or heart disease. However, it is well-known that heart disease is very rare among Japanese because of their diet. This additional information enables Alice to infer that it is highly probable that Umeko is in the hospital because of a viral infection.*

There are more complex forms of background knowledge than the knowledge which can be modeled by negation statements. Consequently, several forms of background knowledge attacks can be launched by the adversaries. To deal with each of such attacks a particular solution is proposed in the literature that we will discuss in the next section.

Skewness attack [61]. This attack occurs when the overall distribution is skewed. The following example illustrates this attack:

Example (skewness attack) [61]*: Consider a dataset containing records of 10000 patients with a set of quasi-identifiers and one sensitive attribute for test result of a virus with two possible values "Positive" and "Negative". Also, assume that test results for 99% of those patients are "Negative" and for just 1% are "Positive". An equivalence class that has equal number of positive and negative cases will be a violation to privacy since each patient belonging to this class will be considered as a positive case with probability of 50% while this probability in the whole dataset is just 1%.*

Similarity attack [61]. When the values of sensitive attribute in an equivalence class are different but semantically similar, the adversaries can attack the privacy of individuals. This attack is called a similarity attack and is shown with the following example:

Example (Similarity attack) [61]*: Consider a dataset with one sensitive value Disease. Assume there is an equivalence class containing three records and the values of attribute Disease for these three records are gastric ulcer, gastritis, and stomach cancer. If Alice knows that Bob's record is in this equivalence class, then, without needing to re-identify Bob's record, she can infer that Bob has a stomach-related disease since all values of attribute Disease in this equivalence class are stomach related. This inference might be a breach of privacy for Bob. This can be also a problem in the case of numerical sensitive attributes. For instance, consider the dataset has also a sensitive attribute Salary and in one of the equivalence classes, containing three records, the values of attribute Salary in those three records are 3K, 4K, and 5K. If Alice knows that Bob's record is in this equivalence class, she can infer that Bob's salary is low since all values of attribute Salary in this equivalence class are in the range [3K-5K].*

Proximity breach [60]. Generally speaking, a privacy breach on a numerical sensitive attribute occurs even if an adversary can infer a *close* value to the exact value of the attribute. This is opposed to privacy violations on categorical attributes that the adversary needs to infer the exact value of sensitive attributes. Besides [61], there are some other works aiming to capture the semantic knowledge of an adversary about a numeric attribute [59, 137]. Li et al. [60] summarized all such privacy breaches on numeric attributes and presented a privacy attack, called *proximity breach*. A proximity breach specifically targets numeric attributes. This privacy attack occurs if the adversary can infer with a high confidence that the numeric sensitive value of an individual falls in a short interval even without knowing the actual value.

Minimality attack [128]. Another type of privacy attack, proposed by Wong et al. [128], is a *minimality attack* which is possible if the adversary has knowledge about the mechanism or algorithm of data anonymization. This attack is based on an implicit principle that all anonymization techniques follow. According to this principle the amount of data distortion in any anonymization process must be always minimum [57] and data modification, using generalization, suppression, etc., should not be done more than nec-

QID	Disease
A	Cancer
A	Cancer
B	Flu
B	Fever
B	Flu
B	Viral infection
B	Diabetes

(a) Original data

QID	Disease
X	Cancer
X	Cancer
X	Flu
X	Fever
X	Flu
X	Viral infection
X	Diabetes

(b) 2-diverse data using global generalization

Name	QID
Jane	A
Steve	A
John	B
Tom	B
Tara	B
Jennifer	B
Mike	B

(c) External public database

Table 3.6: Minimality attack

essary. Based on this minimality principle, an adversary, who knows what mechanism and algorithm employed to anonymize the data, can launch a privacy attack. Wong et al. demonstrated this attack with the following example.

Example (minimality attack) [128]*: Assume the data in Table 3.6a is anonymized such that the number of distinct sensitive values in every equivalence class is at least 2 in order to protect the sensitive values of individuals with a quasi identifier value of A. This is actually the goal of the l-diversity model with $l = 2$ which will be discussed in the next section. This anonymized data is shown in Table 3.6b. Assume an adversary who knows that the l-diversity algorithm is employed to make the data 2-diverse. Also, she knows that a global generalization is used. In addition to this knowledge, the adversary has access to the external data shown in Table 3.6c which is mapped to the same set of individuals in Table 3.6a. Since both quasi identifier values A and B are generalized to a general value X (Table 3.6b), the adversary concludes that there was at least one equivalence class in the original data set violating 2-diversity, because otherwise, according to minimality principle, no generalization was needed. In addition, as in Table 3.6c there are 5 records with a quasi identifier value of B, the adversary will conclude that the*

equivalence class corresponding to B was not violating 2-diversity. Because even if both records with sensitive value Cancer belonged to this group, then there would be 3 other records with non-sensitive values and, therefore, 2-diversity could be satisfied. Based on this reasoning, the adversary concludes that the equivalence class corresponding to A, which contain 2 records, was not satisfying 2-diversity and this will lead her to this conclusion that both individuals with the quasi identifier value A have Cancer, i.e., Jane and Steve.

3.2.3 Privacy Models to Prevent Attribute Disclosure

In this section we review the approaches proposed in the literature to protect the data against attribute disclosure. We classify these methods into three categories based on the type of data these methods are applied to, including relational data, transaction data, and trajectory data.

3.2.3.1 Relational Data

The first privacy model that we evaluate is *l-diversity*. Machanavajjhala et al. [74] showed that there are two types of privacy attacks; namely the *homogeneity attack* and the *background knowledge attack*, that k-anonymity fails to protect against. As a result, the adversary will be able to infer some sensitive information about the individuals even without identifying their records. These facts are employed by Machanavajjhala et al. and they proposed the *l-diversity* principle to overcome the limitations of k-anonymity.

***l*-diversity [74].** A table is l-diverse, if every equivalence class in this table has at least l "well represented" values for the sensitive attribute. A 3-diverse version of the data in Table 3.4 is shown in Table 3.7. It is obvious that this table is not vulnerable to *homogeneity attack* and *background knowledge attack*.

Machanavajjhala et al. [74] presented several instances of the l-diversity principle based on the definition of term "well represented". The simplest model requires that the number of distinct values for sensitive attributes in every equivalence class to be at least

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	≤ 40	*	Heart Disease
4	1305*	≤ 40	*	Viral Infection
9	1305*	≤ 40	*	Cancer
10	1305*	≤ 40	*	Cancer
5	1485*	> 40	*	Cancer
6	1485*	> 40	*	Heart Disease
7	1485*	> 40	*	Viral Infection
8	1485*	> 40	*	Viral Infection
2	1306*	≤ 40	*	Heart Disease
3	1306*	≤ 40	*	Viral Infection
11	1306*	≤ 40	*	Cancer
12	1306*	≤ 40	*	Cancer

Table 3.7: 3-diverse data [74] ©2006 IEEE

l. This is equivalent to the *p-sensitive k-anonymity* principle introduced by Truta and Bindu [119]. Another principle similar to this variant of l-diversity is (α, k)-*anonymity* [126]. A data set is said to satisfy (α, k)-*anonymity* if it satisfies k-anonymity and the probability of inferring sensitive values in every equivalence class is at most α.

Another variant of l-diversity is *entropy l-diversity* that requires the entropy of sensitive attribute in every equivalence class to be at least $\log(l)$ [74]:

$$-\sum_{s \in S} P(EC, s) log(P(EC, s)) \geq \log(l) \tag{3.1}$$

where S is the domain of values of sensitive attribute and p(EC, s) is the fraction of records in equivalence class EC that have value s for the sensitive attribute. For example the entropy of attribute *Condition* in any equivalence class in Table 3.7 is $-\frac{1}{4}\log(\frac{1}{4}) - \frac{1}{4}\log(\frac{1}{4}) - \frac{2}{4}\log(\frac{2}{4}) = log(2.8)$. Therefore this table satisfies entropy 2.8-diversity. This model was first introduced in [87] as a way of protecting against the homogeneity problem without respect to the role of background knowledge [74].

Entropy l-diversity is motivated by the fact that when the frequency of sensitive values

becomes more uniform, then the entropy of sensitive attribute increases [74]. Therefore, by setting a large threshold l, Equation 3.1 will be satisfied if the frequency of a sensitive attribute is close enough to uniform to make the entropy higher than $\log(l)$. However, as the authors in [74] showed, entropy l-diversity will be only possible if the entropy of the sensitive attribute in the entire table is at least $\log(l)$. This constraint, however, may be too restrictive, particularly when a few values of the sensitive attribute are too frequent, for instance, when 90% of patients in a dataset have heart disease [74]. In this case the entropy of sensitive attribute in the entire table will be small and therefore only for a small value of l will entropy l-diversity be satisfied. The other shortcoming of entropy l-diversity is that it cannot be easily adopted to define different levels of protection in the cases that sensitive values have different levels of sensitivity[41].

Another notion of l-diversity is *recursive (c, l)-diversity* [74] which mostly focuses on the role of background knowledge of the adversary. Assuming m_i is the number of sensitive values in the equivalence class i, a dataset satisfies recursive (c,l)-diversity if in every equivalence class i the frequency of the most frequent sensitive value is less than the sum of the frequencies of the $m_i - l + 1$ least frequent sensitive values multiplied by some constant c. That is, if r_j denotes the number of times the j-th most frequent sensitive value appears in equivalence class i and c is a pre-defined constant, equivalence class i satisfies recursive (c,l)-diversity if $r_1 < c(r_l + r_{l+1} + ... + r_{m_i})$. In other words, an equivalence class has recursive (c,l)-diversity if we eliminate one possible value of sensitive attribute and the equivalence class still satisfies (c,l)-diversity [74]. This criterion guarantees that the most frequent sensitive value does not appear too often and the less frequent values do not appear too rarely [41]. However, as it is pointed out in [74], recursive (c,l)-diversity can also be too restrictive.

There is another variant of l-diversity, called *positive disclosure-recursive (c, l)-diversity*, proposed in [74] to deal with the cases that some positive disclosures are acceptable, i.e., when some values of sensitive attribute have less degrees of sensitivity and need not be kept private. The authors define a set Y, called the *don't-care set*, which

contains those sensitive values that have minimal sensitivity and positive disclosure of them is allowed. For example, in a context *flu* may be in set Y but *colon cancer* cannot be. Too frequent sensitive values may also be added to Y, for instance when most of the patients visiting a clinic have heart problems then positive disclosure of value *heart disease* may be allowed by the clinic [74]. Having set Y, data will be anonymized to protect just those sensitive values which are not in Y.

This version of l-diversity addresses two of the criticisms on l-diversity introduced in [61]. The first issue brought up in [61] was that l-diversity may incur an excessive level of anonymization. In other words, there may be some cases where there is no need to enforce l-diversity. To illustrate this problem, assume the dataset of 10000 patients, considered in Section 3.2.2 for skewness attack, i.e., the data set with one sensitive attribute for test result of a virus with two possible values "*Positive*" and "*Negative*". Obviously a negative test result in this case has low sensitivity and a patient will not mind being identified with a negative result. But, on the other hand, a patient with a positive result is very concerned of being known as a positive case. Therefore, achieving *2-diversity* in this dataset will be unnecessary in those equivalence classes which only have "*Negative*" cases. To achieve *2-diversity*, at most $10000 \times 1\% = 100$ equivalence classes from 10000 records can be built and this will obviously lead to a high information loss. *Positive disclosure-recursive (c, l)-diversity* will not anonymize the data unnecessarily in such cases.

The other criticism on l-diversity that is introduced in [61] and can be addressed by *Positive disclosure-recursive (c, l)-diversity* was that l-diversity fails to protect the data against *skewness attack*. Again, consider the dataset of 10000 patients with positive and negative test results. An equivalence class that has equal number of positive and negative cases will satisfy entropy 2-diversity and recursive (c, 2)-diversity. However, this equivalence class is susceptible to a *skewness attack* as was shown in Section 3.2.2. As another example, consider two equivalence classes so that the first one has 49 positive cases and 1 negative and the second one has 49 negative and only 1 positive cases. Both

equivalence classes will be 2-diverse. Also, they will satisfy entropy l-diversity for any $l < 1.05$. However, in the first equivalence class with 49 positive test result, the probability of considering a patient as a positive case is 98% while in the second one this probability is only 2%. But they are dealt with in the same way without considering the issues with skewness of data [61]. On the other hand *positive disclosure-recursive (c, l)-diversity* deal with high sensitive values differently from less sensitive values in the *don't-care* set. Hence, it can easily address the issues with skewed data mentioned above [20].

In addition to the shortcomings discussed above, l-diversity has another problem with respect to the fact that it does not take into account the semantic relations among sensitive values. Therefore, as it was shown in [61], l-diversity fails to protect the data against a *similarity attack*. To overcome the limitations of l-diversity in protecting the data against a *skewness attack* and a *similarity attack*, Li et al. [61] proposed the *t-closeness* privacy model as an extension of l-diversity.

***t*-closeness.** A table has t-closeness if the distance between the distribution of the sensitive attribute in every equivalence class and the whole dataset is at most t. To calculate the distance between distributions the authors use the *Earth Mover Distance(EMD)* metric [97]. This privacy model guarantees that the overall distribution will not be skewed in an equivalence class and therefore a skewness attack cannot be successful. Also, as the distribution of the sensitive attribute in every equivalence class is almost the same as whole dataset, it is unlikely that all values in one equivalence class will be semantically similar. For example, all values of attribute *Disease* in an equivalence class are unlikely to be stomach-related, unless in the case that almost all patients in the dataset have stomach-related diseases.

The limitations of t-closeness are shown in several works. Domingo-Ferrer et al [28] argued that enforcing almost the same distribution for the sensitive attribute in every equivalence class as the whole dataset damages the correlations between quasi identifiers and the sensitive attribute(s), and makes the data useless for analysis [28]. Frikken and

Zhang [40] showed that t-closeness can not deal with the situations where some values of a sensitive attribute has more sensitivity than other values and they prosposed (α_i, β_i)-closeness to address this problem. Their main idea was to assign a range to each sensitive value s_i in the domain of the sensitive attribute. An equivalence class then satisfies (α_i, β_i)-closeness if the number of records in the equivalence class having sensitive value s_i is in the range (α_i, β_i). Another drawback of t-closeness, brought up by Li et al [60], is that *EMD* measure is not an appropriate measure to prevent disclosure of numerical sensitive attributes.

There is a group of works in the literature that specifically address the problem of disclosure of numerical sensitive attributes. One of the privacy principles in this category is *Squared-Error Diversity* [59] proposed by LeFevre et al.

Squared-Error Diversity. This anonymization principle requires that the variance of the numeric sensitive values in every equivalence class to be greater than or equal to a predefined threshold t. However, this privacy principle may not prevent breaches in all cases. For instance, in [60], if in an equivalence class all records except one have the same numeric sensitive value v, according to the squared-error diversity principle, in order to have a large variance on the sensitive values in this equivalence class, the sensitive value of that one record must be sufficiently different from v. However, those records with the same value v are still vulnerable to a privacy breach.

Another principle to guard privacy of numeric sensitive attributes is *(k,e)-anonymity* [137].

(k,e)-anonymity. A dataset is said to satisfy *(k,e)-anonymity* if every equivalence class has at least k different sensitive values and the difference between the maximum and minimum values is at least e. This principle may also suffer from a privacy breach [60]. Assume that we have an equivalence class with k records in which k-1 records have almost identical (but still different) sensitive values, and the remaining record has a very different value with respect to threshold e to satisfy *(k,e)-anonymity*. Those k-1 records will still have a high risk of privacy breach, i.e., $\frac{k-1}{k}$ [60].

Li et al [60] addressed this problem by proposing the (ϵ, m)-*anonymity* principle.

(ϵ, m)-anonymity. This privacy criteria specifically aims to protect the data against the *proximity breach*, introduced in Section 3.2.2. The privacy breaches in [59] and [137] are two examples of a *proximity breach*. This principle is satisfied if for every numerical sensitive value s, the probability of inferring $[s - \epsilon, s + \epsilon]$ is at most $\frac{1}{m}$.

So far, none of the principles we discussed provides a framework to specify the specific privacy requirements of data publisher or data owners. Wang et al [120, 121] proposed such a framework.

Confidence Bounding. Wang et al. [120, 121] proposed a model that enables a data publisher to specify a set of *privacy templates* of the form $\{QID \rightarrow s, h\}$ where QID is a set of quasi identifiers, s is a sensitive value, and h is a threshold. Having the set of privacy templates, the data set satisfies privacy of the individuals if $Conf(QID \rightarrow s) \leq h$ where $Conf(QID \rightarrow s)$ is the maximum confidence of the adversary to infer $QID \rightarrow s$ over all equivalence classes. In other words, the confidence of the adversary in inferring a sensitive value s in an equivalence class is bounded by threshold h. For instance, assume that a data publisher specifies the privacy template $\{\{Job, Age\} \rightarrow HIV, 25\%\}$. This means that the data publisher tolerates no more than 25% confidence for inferring sensitive value *HIV* from quasi identifiers *Job* and *Age*. This privacy template is violated in Table 3.8 because the confidence of inferring *HIV* in equivalence class $\{Accountant, [40, 45)\}$ is 50%.

This model has several advantages [121]. First, the use of privacy templates provides many flexibilities for the data publisher. He can focus only on those parts of the data that have privacy problems and selectively protect only those sensitive values which are exposed to privacy breaches. This will obviously reduce the amount of unnecessary information loss incurred by data anonymization. The data publisher can also specify different thresholds h for different groups of quasi-identifiers and even can specify multiple quasi-identifiers for the same sensitive value. The other advantage is that the confidence measure is a "*user-intuitive*" measure of privacy risk and helps the data publisher to

Job	Age	Disease
Teacher	[35,40)	Viral infection
Teacher	[35,40)	Bronchitis
Teacher	[35,40)	Flu
Teacher	[35,40)	HIV
Accountant	[40,45)	Hepatitis
Accountant	[40,45)	Flu
Accountant	[40,45)	HIV
Accountant	[40,45)	HIV

Table 3.8: 4-anonymous inpatient data

specify the appropriate confidence threshold in an intuitive way. This is an advantage of this model over approaches like entropy l-diversity in which the data publisher may find it difficult to specify a threshold since entropy is not an intuitive measure.

Xiao and Tao [130] proposed a similar approach. However, instead of enforcing a universal privacy requirement on all records, which was assumed in all previous works, they propose to take into account the privacy preferences of every individual whose record is in the dataset.

Personalized Privacy. Xiao and Tao [130] showed that if privacy preferences of data owners are considered and each individual is able to specify his own privacy preferences with respect to his sensitive information, then the amount of information loss will be lower than the approaches with a global privacy requirement like l-diversity. In addition to hierarchies of quasi-identifiers, they considered a hierarchy on the domain of each sensitive attribute and proposed the notion of *personalized privacy*. This model allows individuals to specify nodes in the hierarchy of every sensitive attribute which are sensitive for them with respect to their own privacy preferences. Such nodes are called guarding nodes. The privacy will then be protected if for every guarding node the probability that the adversary can infer a sensitive value in the subtree of that node is less than or equal to a predefined threshold [130].

Similar to [120] and [121], this principle gives more flexibility and reduces the information loss due to data anonymization. However, in practice, it is not easy to obtain privacy preferences of each individual in the dataset, particularly when data had been already collected and after a while it is decided to be released. Moreover, to specify the desired level of privacy, it will be very helpful if the data owner knows the distribution of the sensitive values in the whole data set. For instance if the data owner knows that he has a very common disease like *flu* then he may select a more specific guarding node. But individuals, in many cases, can not have access to such distribution before data release. Therefore, in order to be "on the safe side" data owners will set a high level of privacy protection for their sensitive information that will increase the information loss [41].

In all works discussed so far, the adversary is assumed to know *all* quasi identifiers of victims. However, in real life, when there are many quasi identifiers in a data set, it will be too difficult for an adversary to obtain the information of all quasi identifiers of a victim [82]. Based on this argument, Mohammed et al [82] proposed a privacy principle, called *LKC-privacy*, by assuming that the background knowledge of the adversary is limited to at most L quasi identifiers of an individual.

***LKC*-privacy [82].** A dataset T satisfies LKC-privacy iff for any quasi identifier qid with $|qid| \leq L$

1. $|T[qid]| \geq K$ where $|T[qid]|$ is the number of records in data set T that contain qid and $K > 0$ is the anonymity threshold.

2. $P(s|qid) \leq C$ for any $s \in S$ where S is the set of sensitive values, C is the confidence threshold and $P(s|qid)$ is the probability of inferring sensitive value s knowing qid.

By assuming limited knowledge for the adversary, LKC-privacy model resolves the issues occurring by anonymizing high dimensional data which was brought up by Aggarwal [2] and was discussed in Section 3.1.3.2. LKC-privacy is inspired by the works in [133] and [132] which focus on anonymizing high dimensional transaction data and will be discussed in the next section.

Another type of background knowledge that may lead to attribute disclosure is *probabilistic knowledge* about one part of the domain which was considered by Evfimievski et al [39]. Based on this, the authors proposed the notion of (α, β)-privacy.

(α, β)-privacy. Assume we have an anonymization algorithm R which gets as input a data set with domain D_U and outputs an anonymized data set with domain D_V. For instance, R adds some random noise to input item u and outputs item v. According to the (α, β)-privacy model a privacy breach occurs if the adversary's prior belief that u has property ϕ, i.e., $Pr(\phi(u))$, is very different from adversary's posterior belief, i.e., $Pr(\phi(u)|R(u) = v)$. With this privacy criterion, an adversary's background knowledge is modeled as the prior belief and the additional information she obtains after accessing to anonymized data represents her posterior belief [20]. Evfimievski et al. defined two kinds of privacy breaches, the *upward* (α, β)*-privacy breach* and the *downward* (α, β)*-privacy breach* [39]. Algorithm R leads to an *upward* (α, β)*-privacy breach* with respect to a predicate ϕ if for some probability distribution *f*,

$$\exists u \in D_U, \exists v \in D_V, \; s.t. \quad Pr_f(\phi(u)) \leq \alpha \quad and \quad Pr_f(\phi(u)|R(u) = v) \geq \beta \tag{3.2}$$

Similarly, a *downward* (α, β)*-privacy breach* occurs if:

$$\exists u \in D_U, \exists v \in D_V, \; s.t. \quad Pr_f(\phi(u)) \geq \alpha \quad and \quad Pr_f(\phi(u)|R(u) = v) \leq \beta \tag{3.3}$$

Algorithm R satisfies (α, β)-privacy, if there is no (α, β)-privacy breaches like those defined in Equations 3.2 and 3.3 [39].

3.2.3.2 Transaction Data

All works discussed in previous section focus on anonymizing and publishing a relational data set. In this section we review the methods used to prevent attribute disclosure in transaction data. As we discussed previously, attribute disclosure in transaction data happens when the adversary has knowledge of a subset of a transaction belonging to a target individual, i.e., a subset of items, and applies that knowledge to infer sensitive

	Wine	Strawberries	Meat	Cream	Pregnancy Test	Viagra
Bob	X		X			X
David	X		X			
Claire		X		X	X	
Andrea		X	X			
Ellen	X		X	X		

Figure 3.1: Original transaction data [44] ©2008 IEEE

information. The sensitive information can be either a sensitive item or value of a sensitive attribute which is attached to transaction data.

One of the works in this category is presented by Ghinita et al. [44]. They proposed an anonymization approach for high dimensional transaction data using the bucketization technique [129]. The main idea of bucketization is to partition records into buckets and then de-associate quasi-identifiers from sensitive attributes by randomly permuting the sensitive values in each bucket. The authors assumed a data set in which transactions contain a set of non-sensitive items and a number of sensitive items. An example of such data set is shown in Figure 3.1 [44] where *wine*, *strawberries*, *meat*, and *cream* are non-sensitive items and *pregnancy test* and *viagra* are sensitive. Due to the fact that most transaction data are *sparse*, the authors presented a new representation of data by permuting rows and columns of the original data set such that non-zero values are near the main diagonal of the table. With this representation the adjacent rows (transactions) have a large number of common items and, therefore, have a high correlation. Based on these similarities, transactions are grouped and each group is then associated with a set of sensitive items along with their frequency such that the probability of inferring a sensitive item in each group is lower than a pre-defined threshold. An example of the proposed representation and the anonymized groups are shown in Figures 3.2 and 3.3, respectively. *Bob*, *David*, and *Ellen* are in one group and *Andrea* and *Claire* are in the other one. The probability of inferring that *Bob* or *David* or *Ellen* bought *Viagra* is at most $\frac{1}{3}$ and the probability of buying *pregnancy test* for *Andrea* and *Claire* is at most $\frac{1}{2}$.

One drawback of this method is that the bucketization technique, employed by this approach, can not protect against background knowledge attacks [63]. Moreover, this

	Wine	Meat	Cream	Strawberries	Pregnancy Test	Viagra
Bob	X	X				X
David	X	X				
Ellen	X	X	X			
Andrea		X		X		
Claire			X	X	X	

Figure 3.2: New representation of the transaction data [44] ©2008 IEEE

	Wine	Meat	Cream	Strawberries	Sensitive Items
Bob	X	X			Viagra: 1
David	X	X			
Ellen	X	X	X		
Andrea		X		X	Pregnancy Test: 1
Claire			X	X	

Figure 3.3: Anonymized groups [44] ©2008 IEEE

approach does not make any limitation on the amount of adversary's knowledge and assumes that the adversary knows all non-sensitive items. But this may be an infeasible assumption in practice and might enforce an unnecessary level of protection to the data that will negatively affect the utility of data [132].

Another work to anonymize transaction data is proposed by Xu et al. [133]. Instead of bucketization, the authors introduced an algorithm based on total item suppression for anonymizing data which incurs the minimum amount of information loss. Also, the knowledge of the adversary is assumed to be limited to at most p non-sensitive items in a transaction. Given integers $k > 1$, $p > 0$ and a real $0 < h \leq 1$, Xu et al. [133] assumed an adversary with power p, i.e., an adversary who knows at most p non-sensitive items of a victim, and proposed a new privacy notion, called *(h,k,p)-coherence*.

(h,k,p)-coherence [133]. A data set is said to satisfy (h,k,p)-coherence if, for every non-sensitive itemset β of size at most p, $|\beta| \leq p$, either no transaction contains β or at least k transactions contain β, i.e., $Sup(\beta) = k$, and no more than h percent of these transactions have a common sensitive item [133]. β is called a *mole* if either $Sup(\beta) < k$ or $P_{breach}(\beta) > h$. $P_{breach}(\beta)$ is the maximum probability of inferring any sensitive item through itemset β. A data set is *(h,k,p)-coherent* if it does not contain any moles with respect to p, k, and h.

The challenge is that considering all moles is not practical. This is due to the fact that the number of moles grows exponentially and, therefore, recognizing all moles and eliminating them will not be feasible [133]. In order to address this challenge, Xu et al. consider a smaller set of moles, called "minimal moles" and show that eliminating only minimal moles will lead to data coherence. A mole is minimal if none of its subsets is a mole [133].

In [133], every item is considered as data utility and information loss is measured as the amount of items suppressed. Later Xu et al. [132] extended the work in [133] by considering *nuggets* as data utility. A nugget is an itemset α containing either public or sensitive items and $|\alpha| \leq p'$ and $Sup(\alpha) \geq k'$ where $p' > 0$ and $k' > 1$ [132]. In other words, nuggets are all *frequent* itemsets with respect to the support threshold *k'*. Considering moles and nuggets, the authors' privacy goal is to retain nuggets as much as possible while eliminating all moles. But like [133], due to the exponential growth of moles and nuggets and also the fact that they may have many common items, recognizing all moles and nuggets is not feasible [132]. To deal with this challenge, Xu et al. introduced "maximal" and "minimal" moles and nuggets that form the "borders" for all other moles and nuggets and presented a border based suppression algorithm to make the data (h,k,p)-coherent.

LKC-privacy, discussed in Section 3.2.3.1, employs the idea from these two works to anonymize relational data.

All the works mentioned above have a limitation. They assume that the adversary only knows non-sensitive items and sensitive items are always unknown to the adversary. Cao et al. [18] argued that this assumption may not be true in all real life applications and an adversary may be able to obtain partial knowledge about some sensitive items in a transaction as well.

ρ-uncertainty [18]. Cao et al. assumed an adversary who knows *any* subset X of a transaction t, including sensitive and non-sensitive items, and applies this knowledge to infer a sensitive item $\alpha \notin X$. Authors modeled such an inference as a sensitive asso-

ciation rule (SAR) $X \rightarrow \alpha$ and proposed the notion of *ρ-uncertainty* to anonymize the data such that no SAR can be extracted from the anonymized data. A transaction data set D, satisfies ρ-uncertainty iff for any transaction $t \in D$, any subset of items $X \subset t$ and any sensitive item $\alpha \notin X$, the confidence of any sensitive association rule $X \rightarrow \alpha$ is less than $\rho > 0$. For data anonymization the authors applied a mixed approach that selectively suppresses and generalizes items.

One drawback of this work, like [44], is that the authors assume that the adversary may know itemsets of different sizes and they do not put any limitation on the maximum knowledge of the adversary.

RBAT [70]. Loukides et al. [70], like the works in [68] and [69], focused on specific privacy and utility constraints. They proposed a rule-based anonymization approach, called *RBAT*, to prevent both identity disclosure and attribute disclosure, that enables data owners to formulate their specific privacy requirements in the form of some implication rules called *PS*-rule. These rules has the form $X \rightarrow Y$ which specifies that itemset X should be protected against identity disclosure and sensitive itemset Y should be protected against attribute disclosure through itemset X. Data will be anonymized with respect to these specific privacy constraints while preserving data utility. A top-down algorithm using set-based generalization [68] and [69] is proposed. Later, this work was extended in [71] and in addition to a top-down algorithm, they proposed a sampling based algorithm which uses a combination of top-down and bottom-up generalization heuristics.

3.2.3.3 Trajectory and Sequence Data

A trajectory database may contain sensitive attributes that are associated with trajectory data, such as the attribute *disease*. An adversary who has some background knowledge about the trajectory of a target individual may be able to infer sensitive values of that individual through attribute disclosure. Also, some of the locations visited by individuals may be considered sensitive with respect to the context in which trajectory data is

published, for instance a mental health clinic. An adversary may violate the privacy of a target individual through disclosure of his sensitive location information.

Mohammed et al. [81] define the LKC-privacy model for trajectory data anonymization. LKC-privacy ensures that for every sequence of spatio-temporal pairs with a maximum length L in the trajectory data, there are at least K records, and the ratio of sensitive value(s) in every group is not greater than C. This model prevents both identity disclosure and attribute disclosure, and is achieved by a sequence of global suppression on selected pairs from the trajectory data.

Chen et al. [21] proposed the $(K, C)_L$ privacy model to prevent both identity disclosure and attribute disclosure in trajectory data. They introduced an anonymization algorithm which employs local suppression and global suppression to improve data utility. Their proposed anonymization framework allows the adoption of various data utility metrics for different data mining tasks.

Poulis et al. [92] presented a novel anonymization framework which prevents both identity disclosure and sensitive location disclosure, while preserving data utility. The authors introduced the notion of $(k, l)^m$-*anonymity* which guarantees that an adversary who knows any sub-trajectory s of m nonsensitive locations about an individual, can neither link the individual to fewer than k trajectories in the published dataset, nor can associate the individual with a sensitive location with a probability that exceeds $\frac{1}{l}$. To achieve $(k, l)^m$-anonymity, the distance-based generalization approach [91], which replaces nonsensitive locations with generalized locations, was employed.

In [106], an anonymization framework for anonymizing time-stamped event sequence data is proposed. The authors considered click-stream data corresponding to the browsing history of users in which each event contains a web page visited by a user along with the visit time. They presented a generalization framework to prevent sensitive event disclosure by generalizing time stamps, using time intervals, and events, using a taxonomy which models the domain semantics.

3.3 Discussion

In this chapter we studied the state of the art in privacy preserving data publishing. We categorized approaches into two categories based on types of disclosure and in each category we reviewed potential background knowledge of the adversaries as well as potential privacy attacks based on such knowledge. Several privacy preserving approaches were also discussed in each category for different types of data, i.e., relational data, transaction data, and trajectory data.

Considering the specific properties of longitudinal data which we are studying in this thesis, none of the techniques we reviewed can generate anonymized data which are suitable for emerging longitudinal studies in clinical research. An effective anonymization approach for a longitudinal dataset should consider temporal correlation among the events of each record and should anonymize data such that no combination of values of quasi-identifiers within an event and across events of any record leads to privacy breach, otherwise it cannot provide adequate privacy protection.

Anonymization techniques for relational data and transaction data fail to anonymize longitudinal data as they do not take into account the correlations among multiple records of an individual. Also, they do not consider temporal information in anonymization process.

Although trajectory data anonymization methods consider sequentiality of events in records, they anonymize data from a different perspective, and therefore they cannot be applied to longitudinal data. Moreover, these methods cannot effectively capture all potential privacy attacks in the framework of longitudinal data, because they model an adversary's knowledge as a sequence of locations, times, or (location, time) pairs, whereas in longitudinal data an adversary's knowledge can be in the form of a sequence containing multi-dimensional events. In other words, every event can contain multiple quasi-identifiers' values. For instance, if an adversary knows that Bob had a visit in 2007 while living in ZIP = 56361 and recently was hospitalized for 3 days, this background

knowledge cannot be modeled in the context of trajectory data.

Recently, two methods have been designed to anonymize longitudinal health data [34, 113]. However, both methods only focused on privacy protection against identity disclosure and they fail to prevent privacy attacks through attribute disclosure. Another limitation of these methods is their approach to model background knowledge of adversaries. In longitudinal data, studied in [113], each record contains a sequence of (ICD, Age) pairs as well as a DNA sequence where ICD [1] represents the code of the diagnosis made for a patient and Age is the patient's age at the time of diagnosis. Considering such data, background knowledge of an adversary in this method is modeled as any combination of (ICD, Age) pairs. This method cannot consider the multi-dimensionality of events in our studied longitudinal data, therefore, it cannot model potential background knowledge like "Having a visit in 2007 while living in $ZIP = 56361$ and recently being hospitalized for 3 days". If we want to model our problem in the framework of the method in [113], then we should limit the number of quasi-identifiers in every event to 2 and also we should assume that an adversary always knows the values of both quasi-identifiers in a visit. It is obvious that these assumptions largely impact the effectiveness and efficiency of proposed anonymization techniques for our longitudinal data.

In [34], the authors studied anonymization of a multidimensional longitudinal data and from this perspective their work is related to our studies. However, the authors made two assumptions about the background knowledge of the adversary which makes this work not applicable to our problem. The first assumption is that the adversary would not have any information about co-occurrence of values of quasi-identifiers in one visit. Another assumption is that the adversary would not know the order of visits of a target individual. In other words, in the framework of [34], temporal information available in the longitudinal data is not taken into account for data anonymization. These assumptions limit the work in [34] to provide a complete and effective framework for longitudinal data anonymization due to the fact that it fails to model all potential back-

[1] International Classification of Diseases

ground knowledge of adversaries.

All methods we reviewed in this chapter reside in the category of *syntactic* methods. Syntactic methods achieve anonymity by modifying data through (typically) generalization and suppression until some syntactic condition is met [24]. The goal of these methods is to prevent adversaries from linking their background knowledge about quasi-identifiers to sensitive information. Another category of techniques in the privacy literature is *differential privacy* [32]. In recent years, differential privacy has received a growing interest in data anonymization. The main idea of differential privacy is to add noise to data so that an adversary cannot decide whether a particular record is included in the dataset or not. Since these methods employ noise addition, they cannot preserve data truthfulness which is necessary to preserve in many applications. Moreover, $HIPAA$ clearly specifies a syntactic procedure to anonymize health data, which is longitudinal, in order to meet legal requirements of publishing health data [24]. Therefore, in this thesis, we consider the problem of publishing longitudinal data that simultaneously protects individual privacy under the framework of syntactic models of privacy while preserving data utility.

Chapter 4

HALT: Hybrid Anonymization of Longitudinal Transactions

4.1 Introduction

With recent advances in health informatics and deployment of Electronic Health Records (EHR) and Electronic Medical Records (EMR), vast amounts of Personal Health Information (PHI) are being available. Consequently, demands for accessing and secondary use of such PHI [93] have been increased.

Despite its benefits in various areas, the use of PHI for secondary purposes has increased privacy concerns among the public due to potential privacy risks which can be posed by misuse of health data. Clinical data found in EMRs and EHRs contain information of multiple visits of patients and therefore these data are longitudinal by nature. An example of longitudinal health data is shown in Table 4.1. Every record in this dataset has attributes PID (i.e., patient ID), VID (i.e., visit ID), $AdmYr$ (i.e., admission year), ZIP (ZIP code), $DSFC$ (i.e., number of days since first claim in each year), LOS (i.e length of stay in the hospital), and one sensitive attribute $Disease$. Each patient may have multiple records corresponding to her multiple visits in this dataset and visits of a patient are ordered with respect to visit ID which is assigned based on

PID	VID	AdmYr	ZIP	DSFC	LOS	Disease
1	1	2009	56117	0	3	Hepatitis
2	1	2009	56103	0	2	Infection
3	1	2009	56942	0	1	Fever
3	2	2010	56942	0	30	Cancer
4	1	2009	56107	0	2	Cancer
4	2	2010	56107	0	35	Flu
5	1	2009	56117	0	3	Fever
6	1	2008	56103	0	3	Flu
6	2	2009	56103	0	1	Fever
6	3	2009	56230	40	2	HIV
7	1	2008	56072	0	2	Flu
8	1	2007	56361	0	30	Hepatitis
8	2	2011	56107	0	3	HIV
9	1	2007	56230	0	35	Flu
9	2	2011	56107	0	3	HIV
10	1	2009	56072	0	2	Flu
10	2	2010	56103	0	35	Fever
10	3	2010	56043	100	30	Infection

Table 4.1: Inpatient Longitudinal Data

the visit date. All attributes in this dataset, except the sensitive attribute *Disease* act as QI.

An adversary who has some background knowledge about visits of a target individual is able to launch two kinds of privacy attacks on longitudinal data: *Identity disclosure* and *attribute disclosure*. For instance, if the adversary has access to Table 4.1 and knows that Bob had a visit in 2008 and he has been living in ZIP code 56230 from 2009, then she can uniquely identify Bob's record, #6, and consequently conclude that *Bob* has HIV. This is a case of identity disclosure. In the case of attribute disclosure, for instance, if the adversary knows that Bob had a visit in 2007 and later in 2011 he was hospitalized for 3 days, then both records #8 and #9 match her background knowledge. Although she cannot uniquely identify Bob's record, she can conclude that Bob has HIV with 100% confidence as both records have sensitive value HIV in one of their visits.

In order to prevent these privacy attacks, longitudinal data should be anonymized

such that no combination of values of quasi-identifiers within an event and across all events of any record leads to a privacy breach. Also, the temporal correlation among the events of an individual should be considered in an anonymization process.

In this chapter, we study the challenges in publishing longitudinal health data and propose a new privacy model, together with a data anonymization algorithm to preserve the privacy of patients against both identity disclosure and attribute disclosure. There are some specific requirements [37] for anonymizing health data that must be considered in developing anonymization techniques, otherwise the resulting data will not be practically useful in health care domain. For instance,

- According to the Canadian Institutes for Health Research privacy guidelines [17], for health data anonymization the default approach is to use a hierarchical representation for quasi-identifiers. Therefore, any anonymization algorithm for health data should be able to deal with the hierarchical nature of the quasi-identifiers [37].

- In some data publishing scenarios in health care, it is required that each record in the published data corresponds to a real individual. For instance, a researcher may need the actual records of patients to study the side effects of using a specific drug [41]. This requirement must be considered in developing anonymization algorithms for health data. In particular, when the data is published for general purposes and no specific analysis task is pre-defined for the published data, the anonymization technique needs to meet this requirement in order to make the published data applicable for any kind of analysis. Techniques like *randomization* [6] and *synthetic data generation* [3] fail to satisfy this requirement. Even though cryptographic techniques can preserve the truthfulness of records of patients, the encryption process may alter the semantics required for examining patients' records [41].

In order to satisfy these requirements, we propose an anonymization algorithm using global generalization and global suppression. Using generalization and suppression to-

gether will reduce the overall information loss. This is due to the fact that, on the one hand, suppression can reduce the amount of generalization by removing outliers from the data [65]. On the other hand, instead of completely removing an item from the data, generalization can replace an item with a more general item which is semantically consistent with the original item [110]. Moreover, generalization and suppression, as pointed out in chapter 2, in contrast to the approaches like *swapping* and *randomization* preserve the "truthfulness" of records and, therefore, released data will be meaningful at the record level. This feature is particularly useful for those data mining approaches that are often based on examining records and aggregation information, such as average and frequency, is not of their interest [122]. The other advantage of using generalization and suppression is that they do not add any information to the records which is not in the original data, thus, no fake information will be generated by anonymization.

Despite the benefits of applying global recoding which was discussed in Chapter 2, in general, global recoding incurs high information loss. In order to balance the tradeoff between data privacy and data utility, and decrease the impact of global recoding on the utility of data, we assume that the adversary has *partial knowledge* which is bounded by at most P values of quasi-identifiers. This is a realistic assumption as in real life it may not be feasible for an adversary to acquire the knowledge of all QIs' values in all events in the record of a target individual. We say such adversary has *power* P.

To prevent an adversary with power P from jeopardizing privacy of individuals, we should ensure that adversary cannot find any match for her background knowledge in longitudinal data which can lead her to identity disclosure or attribute disclosure. For this purpose, we define a privacy notion for longitudinal data, called $(K, C)^P$-*privacy*.

Contributions. the major contributions of this chapter can be summarized as follows:

1. We propose a privacy-preserving framework for publishing longitudinal data which prevents both identity disclosure and attribute disclosure attacks. We assume that each record in longitudinal data has a number of quasi-identifiers as well as one sensitive attribute.

2. We assume an adversary who knows at most P values of QIs in the events of a target individual as well as the order of these QIs. Based on this assumption, we propose a privacy notion for longitudinal data, called $(K, C)^P$-*privacy*.

3. We develop an efficient approach to extract privacy breaching sequences in data using pattern discovery techniques [90, 9]. This thesis is the first work which proposed the usage of generalized itemsets and generalized sequences in a privacy preserving approach.

4. We propose a novel anonymization algorithm which anonymizes data by applying global generalization and global suppression.

4.2 Problem Definition

4.2.1 Definitions and notations

In this section we introduce some notations and definitions that are used throughout this chapter. We borrowed some notations from [90, 56]. Let $A = \{A_1, A_2, \ldots, A_n\}$ be a set of attributes and $\Delta = \{\Delta_1, \Delta_2, \ldots, \Delta_n\}$ be the corresponding attribute domains. Each A_i is either a categorical or a numerical attribute. An item is a pair (A_i, $value_i$) which assigns value $value_i \in \Delta_i$ to attribute A_i. An itemset is a set of items. A multidimensional sequence dataset D is a collection of records of the form (SID, S), where SID is a unique ID for every individual and S is an ordered list (sequence) of multidimensional events (itemsets), denoted by $S = \langle e_1, e_2, \ldots, e_m \rangle$. Each multidimensional event e_i is a set of items and contains exactly one item for each attribute in A. A subset of attributes $\{A_1, A_2, \ldots, A_n\}$ is assumed to be publicly available, so they act as quasi-identifiers, $QIs \subseteq \{A_1, A_2, \ldots, A_n\}$. Also, each event can have a number of sensitive attributes which should be kept private. In this study we assume there is only one sensitive attribute in every event. We can trivially extend our work to prevent identity disclosure in the case of multiple sensitive attributes, due to the fact that identity disclosure does not have any

PID	Sequence
1	$\langle\{(AdmYr, 2009), (ZIP, 56117), (DSFC, 0), (LOS, 3), (Disease, Hepatitis)\}\rangle$
2	$\langle\{(AdmYr, 2009), (ZIP, 56103), (DSFC, 0), (LOS, 2), (Disease, Infection)\}\rangle$
3	$\langle\{(AdmYr, 2009), (ZIP, 56942), (DSFC, 0), (LOS, 1), (Disease, Fever)\},$ $\{(AdmYr, 2010), (ZIP, 56942), (DSFC, 0), (LOS, 30), (Disease, Cancer)\}\rangle$
4	$\langle\{(AdmYr, 2009), (ZIP, 56107), (DSFC, 0), (LOS, 2), (Disease, Cancer)\},$ $\{(AdmYr, 2010), (ZIP, 56107), (DSFC, 0), (LOS, 35), (Disease, Flu)\}\rangle$
5	$\langle\{(AdmYr, 2009), (ZIP, 56117), (DSFC, 0), (LOS, 3), (Disease, Fever)\}\rangle$
6	$\langle\{(AdmYr, 2008), (ZIP, 56103), (DSFC, 0), (LOS, 3), (Disease, Flu)\},$ $\{(AdmYr, 2009), (ZIP, 56103), (DSFC, 0), (LOS, 1), (Disease, Fever)\},$ $\{(AdmYr, 2009), (ZIP, 56230), (DSFC, 40), (LOS, 2), (Disease, HIV)\}\rangle$
7	$\langle\{(AdmYr, 2008), (ZIP, 56072), (DSFC, 0), (LOS, 2), (Disease, Flu)\}\rangle$
8	$\langle\{(AdmYr, 2007), (ZIP, 56361), (DSFC, 0), (LOS, 30), (Disease, Hepatitis)\},$ $\{(AdmYr, 2011), (ZIP, 56107), (DSFC, 0), (LOS, 3), (Disease, HIV)\}\rangle$
9	$\langle\{(AdmYr, 2007), (ZIP, 56230), (DSFC, 0), (LOS, 35), (Disease, Flu)\},$ $\{(AdmYr, 2011), (ZIP, 56107), (DSFC, 0), (LOS, 3), (Disease, HIV)\}\rangle$
10	$\langle\{(AdmYr, 2009), (ZIP, 56072), (DSFC, 0), (LOS, 2), (Disease, Flu)\},$ $\{(AdmYr, 2010), (ZIP, 56103), (DSFC, 0), (LOS, 35), (Disease, Fever)\},$ $\{(AdmYr, 2010), (ZIP, 56043), (DSFC, 100), (LOS, 30), (Disease, Infection)\}\rangle$

Table 4.2: Converted Inpatient Longitudinal Data

condition on sensitive attributes. However, this extension will not be so trivial for the case of attribute disclosure. More precisely, to prevent attribute disclosure we cannot consider each sensitive attribute separately and we need to make some changes in our proposed approach to consider combinations of different sensitive attributes' values. The extension to multiple sensitive attributes will be handled in our future work. The other attributes which are neither QI nor sensitive do not have any impact on data anonymization and can be ignored during the anonymization process. Consider inpatient data in Table 4.1. By expressing each visit as a set of $(A_i,\ value_i)$ pairs, $i \in 1, \ldots, n$, and grouping all visits of every patient together, sorted by visit dates, longitudinal data in Table 4.1 can be represented as data in Table 4.2. In the rest of this chapter, we use the dataset in Table 4.2 as a running example in our discussion. The size m of a sequence S, denoted by $|S|$, is the number of events (itemsets) in the sequence. The length of a sequence $S = \langle e_1, e_2, \ldots, e_m\rangle$, denoted by $len(S)$, is defined as the total number of items in the sequence,

i.e., $len(S) = |S| \cdot n$. A sequence with length ℓ is called an ℓ-sequence. For instance, record PID#8 in Table 4.2 is a sequence with size 2 and length 10. A sequence $S' = \langle e'_1, e'_2, \ldots, e'_{m'} \rangle$ contains another sequence $S = \langle e_1, e_2, \ldots, e_m \rangle$, denoted by $S \subseteq S'$, if there are integers $1 \leq j_1 < j_2 < \cdots < j_m \leq m'$ such that $e_1 \subseteq e'_{j_1}$, $e_2 \subseteq e'_{j_2}$, $\ldots$, $e_m \subseteq e'_{j_m}$. S is called a subsequence of S' and S' is a supersequence of S [9]. For example, sequence $\langle \{(AdmYr, 2009), (DSFC, 0)\}, \{(AdmYr, 2010)\} \rangle$ is a subsequence of the sequence #4. We assume that each quasi-identifier attribute $q \in QI$, is associated with a generalization hierarchy denoted by H_q. Figure 4.1 shows generalization hierarchies of the QIs of data in Table 4.1. In the presence of generalization hierarchies, quasi-identifiers can get values from any level of their corresponding hierarchy. In other words, items in the events of a sequence, i.e., (A, value) pairs, can be expressed in different levels of granularity [90].

Definition 4.1. (Generalized Item). Given a generalization hierarchy H_q for quasi-identifier q, every pair $(q, value_q)$ is a generalized item, or g-item in short [56].

We can compare g-items using a specificity relation, defined as follows.

Definition 4.2. (Item Specificity Relation). For every generalized items $x = (q, v)$ and $y = (q, v')$, y is said to be more specific than x, denoted by $x \leq_I y$, if $v = v'$ or $v = \hat{v'}$, where $\hat{v'}$ denotes an ancestor of v' with respect to H_q. Relation $\leq_I$ is a partial ordering over the set of all generalized items. In other words, this relation over D is reflexive, anti-symmetric, and transitive.

Definition 4.3. (Generalized Itemset). The generalized itemset (g-itemset) $e = \{x_1, \ldots, x_t\}$ is a non-empty set of generalized items such that for all distinct i and j in $\{1, \ldots, t\}$, x_i and x_j are not in an ancestor-descendant relationship with respect to their corresponding generalization hierarchy.

Definition 4.4. (Generalized Sequence). The generalized sequence (g-sequence) $S = \langle e_1, \ldots e_z \rangle$ is a non-empty ordered list of generalized itemsets e_i, $i \in \{1, .., z\}$. We assume that all g-items, corresponding to a specific quasi-identifier q, in different itemsets of a g-sequence are at the same level with respect to the q's generalization hierarchy.

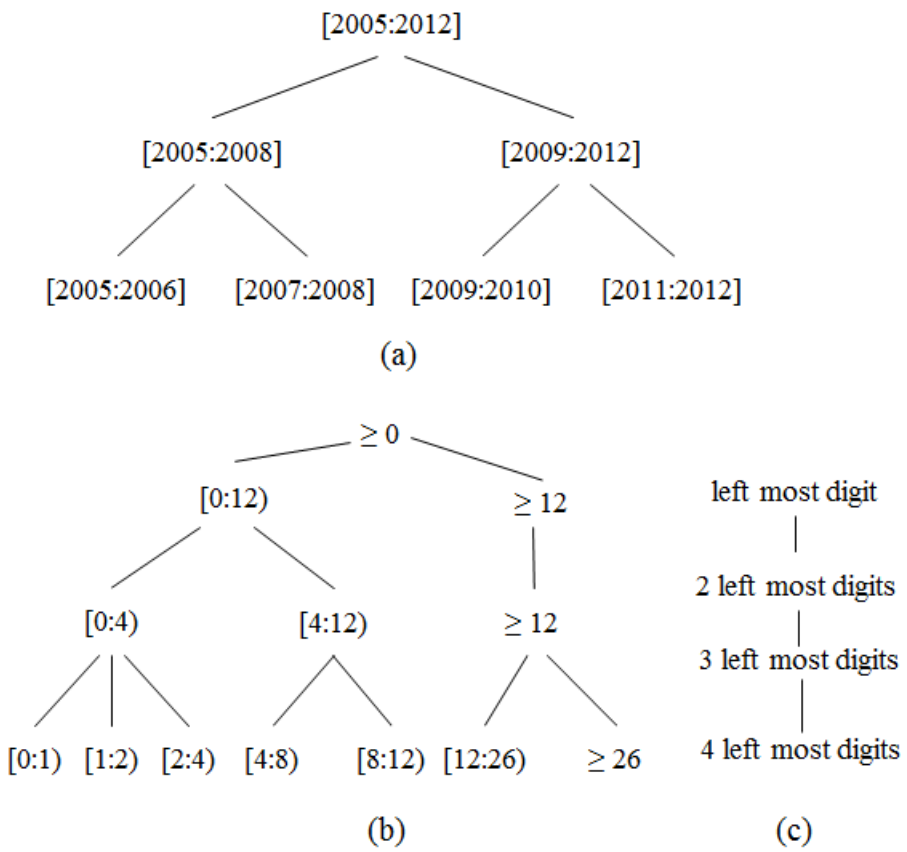


Figure 4.1: Generalization hierarchy for (a) *AdmYr* (b) *DSFC* and *LOS* in terms of number of weeks (c) *ZIP*

We can extend specificity relation to generalized itemsets and generalized sequences as follows.

Definition 4.5. (Itemset Specificity Relation). Let $e = \{x_1, \ldots, x_t\}$ and $e' = \{x'_1, \ldots, x'_{t'}\}$ be two g-itemsets. G-itemset e' is said to be more specific than e (denoted by $e \leq_{IS} e'$), if for every x_i in e, there exists $x'_{i'}$ in e' such that $x_i \leq_I x'_{i'}$.

For instance, $\{(AdmYr, 2009), (LOS, [0 : 4))\} \leq_{IS} \{(AdmYr, 2009), (ZIP, 56072), (LOS, 2)\}$. Relation $\leq_{IS}$ generalizes set inclusion in the sense that for all itemsets e and e', such that $e \subseteq e'$, then $e \leq_{IS} e'$.

Definition 4.6. (Support of a Generalized Itemset). The support of a generalized

itemset $e = \{x_1, \ldots, x_t\}$, denoted by $sup(e)$, is the number of records in the multidimensional sequence dataset D which contain g-itemset $e' = \{x'_1, \ldots, x'_{t'}\}$ such that $e \leq_{IS} e'$.

For instance, $sup(\{(AdmYr, 2009), (LOS, [0:4))\})$ is 7.

Proposition 4.1. *For all g-itemsets e and e', if $e \leq_{IS} e'$, then $sup(e') \leq sup(e)$.*

Proof. Refer to [56]. □

Definition 4.7. (Sequence Specificity Relation). Given g-sequences $S = \langle e_1, \ldots, e_m \rangle$ and $S' = \langle e'_1, \ldots, e'_{m'} \rangle$, S' is said to be more specific than S, denoted by $S \leq_S S'$, if there are integers $1 \leq j_1 < j_2 < \cdots < j_m \leq m'$ such that $e_1 \leq_{IS} e'_{j_1}, e_2 \leq_{IS} e'_{j_2}, \ldots, e_m \leq_{IS} e'_{j_m}$.

Given two g-sequences S and S', if $S \leq_S S'$, then we say that sequence S' contains sequence S.

Definition 4.8. (Support of a Generalized Sequence). The support of a generalized sequence S is the number of records in the sequence data which contain S.

Proposition 4.2. *For all g-sequences S and S', if $S \leq_S S'$, then $sup(S') \leq sup(S)$.*

Proof. Refer to [56]. □

4.2.2 Privacy Model

An important factor in developing effective privacy models is to make reasonable assumptions about the background knowledge of adversaries. If the knowledge of the adversary is underestimated, this will lead to an insufficient level of privacy protection. On the other hand, overestimating an adversary's knowledge will result an unnecessary level of protection which may cause high data distortion. We assume an adversary who knows that the record of a target individual exists in a released longitudinal dataset and her goal is to obtain new (sensitive) information from data about that individual. Moreover, the adversary is assumed to have *some* background knowledge about the longitudinal events of a target individual, i.e., values of QIs as well as the order of these values in the

events of an individual's record. In the presence of generalization hierarchies for QIs, which are publicly available, the adversary can express her knowledge of QIs values at different levels of granularity. For instance, the adversary may know that the target individual had a visit in 2005 and, using the generalization hierarchy of $AdmYr$ given in Figure 4.1, she can represent her knowledge as ($AdmYr$, [2005:2006]); or the adversary may know that the target individual had a visit between 2005 and 2008 and she can represent this knowledge as ($AdmYr$, [2005:2008]).

As we mentioned, in this chapter our goal is to propose a practical anonymization technique which can satisfy specific requirements of anonymizing longitudinal health data. Therefore, we opted to apply global recoding where all records have the same recoding within each quasi-identifier. More precisely, if a quasi-identifier q should be generalized, all its values should be generalized to their ancestor values at the same level of the corresponding generalization hierarchy H_q. Also, all instances of a value of q should be suppressed if it is a candidate for suppression. Global recoding ensures that there is no inconsistency in data analysis results obtained from anonymized data compared to the original data. However, the information loss incurred by global recoding is higher than local recoding. On the other hand, since the main purpose of publishing data is to provide rich sources of information for data analysis tasks, preserving data utility is of high importance. To balance the tradeoff between data privacy and data utility, and decrease the impact of global recoding on the utility of data, we assume that the adversary has *partial knowledge* which is bounded by at most P values of quasi-identifiers, i.e., P g-items. This is a realistic assumption as in real life it may not be feasible for an adversary to acquire the knowledge of all QIs' values in all events.

There are several possible combinations of P g-items with respect to their order and co-occurrence in the events of an individual's record. We can model such background knowledge by a generalized sequence $X = \langle e_1, \ldots, e_j \rangle$ where $len(X) \leq P$. For example, if the adversary knows that Bob had a visit in 2008 and also hospitalized for 2 days in 2009, she knows values of 3 generalized items two of which occur together in one visit. Such

knowledge can be modeled as $X = \langle\{(AdmYr, 2008)\}, \{(AdmYr, 2009), (LOS, 2)\}\rangle$.

Armed with such background knowledge, the adversary can launch a privacy attack by finding some matching records to her background knowledge X in the longitudinal dataset. A record matches a g-sequence X if it contains X. For example, considering data in Table 4.2, records with PIDs #8 and #9 match sequence $X = \langle\{(AdmYr, 2007)\}, \{(AdmYr, 2011), (LOS, [0:4)\}\rangle$.

The adversary's background knowledge X can lead to a privacy breach if:

a. the number of records containing X, i.e., $sup(X)$, is less than a support threshold k, i.e., $sup(X) < k$; or

b. there are more than k matching records to X, but the percentage of records, among these k records, which have a high sensitive item s in one of their events is greater than a threshold $c \in (0, 1]$. Such inference can be modeled by the inference rule $X \rightarrow s$ where $conf(X \rightarrow s) = \frac{sup(X \cap s)}{sup(X)} > c$.

Any generalized sequence X that leads to the inference of new (sensitive) information about a target individual is an *inference channel*.

Definition 4.9. (Inference Channel). Given an anonymity threshold k and a confidence threshold c, generalized sequence $X = \langle e_1, \ldots, e_j\rangle$ where $len(X) \leq P$, is an inference channel, IC in short, if $sup(X) < k$ or $conf(X \rightarrow s) > c$ for any high sensitive value s.

As mentioned in proposition 4.2, for all g-sequences S and S', if $S \leq_S S'$, then $sup(S') \leq sup(S)$. This implies that if the support of a generalized sequence S is less than k, meaning S is an inference channel, then all generalized sequences S' such that $S \leq_S S'$ are also inference channels. However, this property does not hold for the confidence of inference rules. In other words, if the confidence of inferring a high sensitive value s through a generalized sequence X' is less than c, then we cannot say that confidence of generalized sequence X, such that $X \leq_S X'$, is also less than c. This means

that a generalized sequence X can be an inference channel but none of the generalized sequences X', such that $X \leq_S X'$, are inference channels.

Observation 4.1. *A generalized sequence $X = \langle e_1, \ldots, e_j \rangle$ may be an inference channel, even if none of its specializations is an inference channel.*

This observation is illustrated in the following example.

Example 4.1. Let $k = 2$, $c = 0.5$, and $P = 2$. Also, assume that $Cancer$ and HIV are high sensitive values for the attribute Disease. Let $X = \langle \{(AdmYr, 2009)\}, \{(LOS, 30)\} \rangle$ and $X' = \langle \{(AdmYr, 2009)\}, \{(LOS, 35)\} \rangle$. None of these sequences is an inference channel in table 4.2, because

$$Sup(X) = 2,\ conf(X \rightarrow Cancer) = 0.5, \text{ and } conf(X \rightarrow HIV) = 0$$

$$Sup(X') = 2,\ conf(X' \rightarrow Cancer) = 0.5, \text{ and } conf(X' \rightarrow HIV) = 0$$

With respect to the generalization hierarchy of LOS in Figure 4.1, we can generate the g-sequence $Y = \langle \{(AdmYr, 2009)\}, \{(LOS, [4:8))\} \rangle$ which is more general than both sequences X and X', i.e., $Y \leq_S X$ and $Y \leq_S X'$ (recall that in LOS's generalization hierarchy non-leaf nodes are represented in terms of number of weeks). If we calculate the support and confidence of g-sequence Y, we will find that it is an inference channel because $conf(Y \rightarrow Cancer) = 0.6 > C = 0.5$.

To prevent an adversary from jeopardizing privacy of individuals, we should detect and remove all inference channels contained in longitudinal data before release. This ensures that the adversary cannot find any match for her background knowledge in the data. For this purpose, we define the following privacy notion.

Definition 4.10. ($(K, C)^P$-privacy). Given support threshold $K \geq 2$, confidence threshold $C \in (0, 1]$, and power threshold $P \geq 1$, a longitudinal dataset D satisfies $(K, C)^P$-privacy if D does not contain any generalized sequence X with $len(X) \leq P$, such that $sup(X) < K$ or $conf(X \rightarrow s) > C$, for any high sensitive value s in the domain

of the sensitive attribute. In other words, $(K, C)^P$-privacy ensures that, for an adversary with power P, the probability of linking an individual to a sequence is at most $\frac{1}{K}$ and the probability of linking an individual to a high sensitive item is at most C.

4.2.3 Information Loss

Data anonymization inevitably leads to some information loss. We consider the scenario where the data analysis task is unknown at the time of publication. Therefore, the overall data distortion should be measured as the amount of information loss incurred by data anonymization. To enforce $(K, C)^P$-privacy, first we identify all inference channels in the data and then we employ global generalization combined with global suppression to eliminate detected inference channels. Our anonymization approach is represented with two sets Λ and Φ which stand for generalization strategy and suppression strategy, respectively. The generalization strategy determines the appropriate level of generalization for every quasi-identifier in QI and the suppression strategy identifies a set of QIs' values, i.e., g-items, for suppression.

The set of all possible generalization strategies for QIs form a generalization lattice [74]. Each node Λ_i in this lattice corresponds to a unique generalization strategy and defines a possible anonymized instance of the dataset. These nodes are partially ordered by a generalization relation $<_G$. $\Lambda_i <_G \Lambda_j$ indicates that the values in the generalization defined by Λ_j are the generalizations of the values in the generalization defined by Λ_i. The bottom-most node in this lattice corresponds to the case NO-GENERALIZATION and the top most node corresponds to the case ANY which means all quasi-identifiers are generalized to the root of their generalization hierarchy. We search this lattice in a top-down fashion to find an anonymization solution which eliminates all inference channels while preserving data utility as much as possible. At each node, we check if its corresponding generalization strategy can eliminate the extracted inference channels. If there are some inference channels that cannot be eliminated by that generalization strategy, we find a global suppression strategy to efficiently remove those inference channels.

When the total information loss cannot be decreased anymore, our algorithm terminates.

To capture the amount of the information loss incurred by global generalization and global suppression, we define an information loss metric similar to *Loss Metric* (LM) [50]. Given a generalization strategy Λ and the hierarchy of the quasi-identifier q, the information loss at node g in this hierarchy is defined as:

$$InfoLoss_{\Lambda}(q, g) = \frac{|leaves(q, g)|}{|Domain(q)|} \tag{4.1}$$

where $|leaves(q, g)|$ is the number of leaves in the subtree rooted at g in the hierarchy of quasi-identifier q and $|Domain(q)|$ is the domain size of quasi-identifier q, i.e., the number of leaves in the generalization hierarchy of q. If g is a leaf node then $|leaves(q, g)|$ will be 0. For example, information loss at node [2005 : 2008] in the generalization hierarchy of $AdmYr$ is $\frac{4}{8} = 0.5$, since node [2005 : 2008] has 4 leaf values including 2005, 2006, 2007, and 2008 and the domain size of $AdmYr$ is 8, including 2005, 2006, 2007, 2008, 2009, 2010, 2011, and 2012.

$InfoLoss_{\Lambda}(q, g)$ gives the information loss of generalizing one instance of a leaf value in the subtree rooted at node g to g. Therefore, the total information loss of generalizing all instances of values in the leaves of g's subtree to g is

$$InfoLoss^{D}_{\Lambda}(q, g) = \sum_{n \in leaves(q,g)} count(n) * InfoLoss_{\Lambda}(q, g) \tag{4.2}$$

where $count(n)$ is the number of instances of the value n in the dataset D.

Since we apply global generalization, all instances of values of a quasi-identifier should be generalized to their corresponding ancestor values in the hierarchy level indicated by the generalization strategy Λ. Therefore, the total information loss of generalizing values of quasi-identifier q with their ancestor nodes at level l is

$$InfoLoss^{D}_{\Lambda}(q) = \sum_{g \in l} InfoLoss^{D}_{\Lambda}(q, g) \tag{4.3}$$

Finally, the total information loss incurred by the generalization strategy Λ in the whole dataset can be defined as follows:

$$InfoLoss_D(\Lambda) = \sum_{q \in QI} InfoLoss^D_\Lambda(q) \tag{4.4}$$

For calculating the information loss incurred by the suppression strategy Φ, we should calculate the information loss of suppressing every quasi-identifier value, i.e., generalized item, in Φ. It is important to note that based on the level of generalization for every quasi-identifier in the generalization strategy Λ, quasi-identifiers' values in Φ can be either a leaf value or a generalized value. For instance, assume that generalization levels for the quasi-identifiers LOS and $AdmYr$ in a generalization strategy Λ are 0 and 1, respectively. Therefore, in the anonymized dataset, based on this generalization strategy, values of LOS are leaf values of the hierarchy of LOS, e.g. 2 and 4, while values of $AdmYr$ are values of the nodes at level 1 of the generalization hierarchy of $AdmYr$, such as [2005:2006] and [2009:2010].

Given the suppression strategy Φ, for every generalized item $\phi \in \Phi$, if ϕ corresponds to a leaf value, the information loss of suppressing one instance of ϕ is 1. Therefore, the total information loss for suppressing every such ϕ will be $count(\phi)$. On the other hand, if ϕ corresponds to a non-leaf value in its generalization hierarchy, then the information loss of suppressing one instance of ϕ is not equal to 1, since some information has been already lost due to generalization. More precisely, all values in the leaves of subtree rooted at ϕ have been generalized to ϕ according to the generalization strategy. Therefore, the information loss of suppressing one instance of such ϕ is

$$1 - InfoLoss_\Lambda(\phi_q, \phi_{val}) \tag{4.5}$$

where ϕ_q is the corresponding quasi-identifier of generalized item ϕ, and ϕ_{val} is the value of the quasi-identifier.

Thus the total information loss of suppressing every generalized item ϕ, either leaf or non-leaf, in dataset D can be expressed as:

$$InfoLoss^D_\Phi(\phi) = count(\phi) * (1 - InfoLoss_\Lambda(\phi_q, \phi_{val})) \tag{4.6}$$

The above equation defines information loss for every generalized item ϕ in the suppression strategy Φ. So, the total information loss due to suppression strategy Φ in the whole dataset D will be:

$$InfoLoss_D(\Phi) = \sum_{\phi \in \Phi} InfoLoss_{\Phi}^{D}(\phi) \tag{4.7}$$

Having information loss incurred by generalization strategy Λ and suppression strategy Φ, the normalized total information loss for every possible anonymization solution will be

$$InfoLoss_D(\Lambda, \Phi) = \frac{InfoLoss_D(\Lambda) + InfoLoss_D(\Phi)}{|D| \times |QI|} \tag{4.8}$$

4.2.4 Problem Statement

To satisfy $(K, C)^P$-privacy requirement, we employ global generalization along with global suppression techniques to modify data set D. However, data modification reduces the utility of data. Therefore, among several possible solutions, the anonymization algorithm should find the optimal solution which has the minimum information loss and preserves data utility as much as possible. However, finding an optimal solution for this problem is NP-hard. This can be proved by converting the optimal anonymization solution into the *vertexcoverproblem* [25]. So, we employ some heuristics to efficiently find a sub-optimal solution, with respect to the $(K, C)^P$-privacy model. The problem we are addressing is summarized as follows:

Problem 4.1. Given longitudinal data D, anonymity threshold K, confidence threshold C, adversary's power P, and the generalization hierarchies of quasi-identifiers, the longitudinal data anonymization problem is to transform data set D to an anonymized data set D', using global generalization and global suppression, to satisfy $(K, C)^P$-privacy while minimizing information loss defined by Equation 4.8.

4.3 Anonymization Framework

As it is defined in Definition 4.9, every generalized sequence X, such that $len(X) \leq P$, which leads to identity disclosure or attribute disclosure is an inference channel. So, if we remove all inference channels, the anonymized data will satisfy $(K, C)^P$-privacy and will be protected against privacy attacks. We propose an anonymization framework, called *Hybrid Anonymization of Longitudinal Transactions* ($HALT$), which has two steps. At first, we extract inference channels in the longitudinal data. Then we greedily search a *multi-domain generalization lattice* [37, 57] in a top-down, breadth-first manner to find a hybrid anonymization solution, using generalization and suppression, to eliminate extracted inference channels, while maintaining as much data utility as possible.

4.3.1 Step 1: Extracting Inference Channels

As we explained, due to the *anti-monotonicity* property of support of a generalized sequence X, if $sup(X) < K$ then for every generalized sequence X' with length less than or equal to P, such that $X <_S X'$, $sup(X') < K$. This means that if a generalized sequence X is an inference channel through its support, then all sequences which are more specific than X are also inference channels. This implies that the number of inference channels may grow dramatically, and extracting all inference channels would not be practical. To address this challenge, we consider a smaller set of inference channels, called *minimal inference channels*, and show that it suffices to *only* extract and eliminate minimal inference channels in order to satisfy $(K, C)^P$-privacy.

Definition 4.11. (Minimal Inference Channel). A generalized sequence X with length less than or equal to P is a minimal inference channel (MIC), if X is an inference channel and there is no inference channel X' which is more general than X. More precisely, the set of all minimal inference channels w.r.t. support threshold $K \geq 2$, confidence threshold $C \leq (0, 1]$, power threshold $P \geq 1$, and generalization hierarchies of quasi-identifiers, is the set Ω of generalized sequences with the following properties:

$\forall X \in \Omega$

- $len(X) \leq P$
- $sup(X) < K$ or $conf(X \rightarrow s) > C$, for every high sensitive value s
- $\forall X' \in \Omega$: $X' \neq X$, $sup(X') < K$ or $conf(X' \rightarrow s) > C \Longrightarrow \neg(X' <_S X)$

The following theorem indicates that in order to make a dataset satisfy $(K, C)^P$-privacy, it is enough to eliminate minimal inference channels.

Theorem 4.1. *A dataset D satisfies $(K, C)^P$-privacy if and only if D contains no minimal inference channels.*

Proof. Assume dataset D does not contain any minimal inference channel, but it still does not satisfy $(K, C)^P$-privacy. Based on definition of $(K, C)^P$-privacy, D must contain some inference channels and, by Definition 4.11, these inference channels should be either minimal or more specific than a minimal inference channel. But this is a contradiction to the initial assumption and therefore, D satisfies $(K, C)^P$-privacy. □

So, in the first step we extract all minimal inference channels from longitudinal data. We propose an efficient algorithm to identify all generalized sequences which are minimal inference channels. A good strategy to efficiently identify minimal inference channels is to enumerate every more general g-sequence before all its more specific g-sequences. In this way, we can prune the search space by avoiding generating inference channels which are not minimal. For this purpose, we assume an order for all g-items, corresponding to a specific quasi-identifier q, based on the depth-first order of nodes in the generalization hierarchy of q. We also impose an order for all quasi-identifiers. As a result all g-items corresponding to the quasi-identifier i have higher order than all g-items corresponding to the quasi-identifier j, for $i < j$, and for all g-items corresponding to a specific quasi-identifier q, any non-leaf g-item will have higher order than all its descendants in the q's generalization hierarchy. We assign item IDs to the g-items according to the proposed

order, so every g-item has a smaller item ID than all its descendants.

We build a lexicographic tree, called *lexicographic g-sequence tree*, to enumerate and organize generalized sequences. Our approach for enumerating sequences is inspired from the $SPAM$ algorithm [9], which is an efficient algorithm for sequential pattern mining [73]. Each node in the lexicographic g-sequence tree is a g-sequence and all g-sequences are organized in the tree based on the *sequence specificity relation*, defined in Definition 4.7. More precisely, every node n at level l in this tree corresponds to a g-sequence X of length l. All g-sequences X' of length l, such that $X <_S X'$, are on the right side of X at the same level and g-sequences X'' of length $l+1$, such that $X <_S X''$, are child nodes of X.

We first sort all g-items in ascending order of their ids. Then, in an initial step, for every g-item Z, if it is an inference channel (a g-item can be seen as a g-sequence of length 1), we add it to the set Ω of extracted minimal inference channels and eliminate all g-items Z' which are more specific than Z, i.e., $Z <_I Z'$, based on the fact that none of them can be a minimal inference channel. The proposed ordering of g-items ensures that we always evaluate a more general g-item before all its descendants. After finding all minimal inference channels of length 1, remaining g-items are non-inference channels of length 1 which will be generators of all the other g-sequences. In the next step, we add all these non-inference channel g-items as g-sequences of length 1 to the g-sequence tree. We traverse the g-sequence tree in a top-down, breadth-first manner and at each node n we generate candidate child g-sequences of length $l+1$ by extending the g-sequence of length l in node n.

Every g-sequence S of length l in the tree can be extended by joining with another g-sequence S' of the same length, through two different procedures:

- **sequence-extension:** In a sequence-extension approach, a child g-sequence is generated by appending a new g-itemset consisting of a single g-item to the end of the parent g-sequence S. The new g-itemset can be added only if it is not a descendant or ancestor of any g-item in S. We refer to this process as the *S-step*. For example,

if we have g-sequence $X = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}\rangle$, then $X' = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}, \{(LOS, 30)\}\rangle$ is a sequence-extended g-sequence.

- **itemset-extension:** In an itemset-extension approach, a child g-sequence is generated by adding a single g-item i to the last g-itemset of the parent g-sequence S. Item i can be added to the last g-itemset e only if a) i's ID is greater than any g-item's ID in e, b) there is no other g-item in e corresponding to the same quasi-identifier as i, c) i is not a descendant or ancestor of any g-item in S. We refer to this process as *I-step*. For example, if we have g-sequence $X = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}\rangle$, $X'' = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (LOS, 30)\}\}\rangle$ is an itemset-extended g-sequence.

Two g-sequences S and S' can be joined if they have the same prefix. More precisely, if $S = \langle\{x_{11}, x_{12}, \ldots, x_{1t}\}, \ldots, \{x_{m1}, x_{m2}, \ldots, x_{m(t-1)}, x_{mt}\}\rangle$ and $S' = \langle\{x_{11}, x_{12}, \ldots, x_{1t}\}, \ldots, \{x_{m1}, x_{m2}, \ldots, x_{m(t-1)}, x'_{mt}\}\rangle$, then they can be joined through S-step and I-step. The resulting g-sequence of S-step will be $\langle\{x_{11}, x_{12}, \ldots, x_{1t}\}, \ldots, \{x_{m1}, x_{m2}, \ldots, x_{m(t-1)}, x_{mt}\}, \{x'_{mt}\}\rangle$, and the resulting g-sequence of I-step will be $\langle\{x_{11}, x_{12}, \ldots, x_{1t}\}, \ldots, \{x_{m1}, x_{m2}, \ldots, x_{m(t-1)}, x_{mt}, x'_{mt}\}\rangle$. The join condition ensures that we do not generate multiple instances of the same potential candidate g-sequence.

For every candidate g-sequence X, if it is an inference channel, we stop extending X and if X is not a specialization of any previously extracted MIC, we add X as a new MIC to the set of minimal inference channels Ω. If X is not an inference channel, we add X to the list of NIC g-sequences for the next round. After enumerating all MICs with length less than or equal to P the algorithm terminates.

In order to reduce the number of candidate g-sequences which can be generated from a g-sequence X in S-step and I-step, we associate two sets with every g-sequence X in the tree: $lock_S$, the set of all g-items that cannot be joined to X in an S-step extension, and $lock_I$, the set of all g-items that cannot be joined to X in an I-step extension. These

sets contain g-items which if joined with X will generate g-sequences which are more specific than one of the already identified minimal inference channels. Such g-sequences cannot generate any new minimal inference channels and therefore we prune them. $lock_S$ and $lock_I$ also contain all descendants and ancestors of g-items in the g-sequence X. This is with respect to the assumption we made about the generalization level of g-items in a g-sequence, as specified in Definition 4.4. All g-items in the sets $lock_S$ and $lock_I$ of every g-sequence X are also added to the sets $lock_S$ and $lock_I$ of all more specific g-sequences generated from X.

One of the most expensive operations in extracting minimal inference channels is to count the support of each candidate g-sequence. A naive approach would be to scan the dataset each time we need to count the support. But this is not practical, because scanning is a time consuming operation. So, we need to find a more efficient technique.

Counting support of itemsets and sequences has been essentially a challenge in the literature of *frequent pattern mining* [47]. Among several techniques which have been proposed to address this issue, vertical representation of datasets has been a widely used approach [9, 15, 135, 136]. In order to efficiently perform the support counting, we employ a *vertical bitmap representation* of longitudinal data inspired from [9]. By performing bitwise operations on the bitmaps, the support of g-sequences can be obtained quickly.

For every g-item i in the longitudinal data, we construct a vertical bitmap. For each sequence in the data, there is a partition in the bitmap of a g-item corresponding to its events. If event e contains g-item i, the corresponding bit in the bitmap of i is set to 1; otherwise the bit is set to 0. Having bitmaps of all g-items, we can easily and efficiently calculate bitmaps of g-itemsets and g-sequences. Assume we have bitmaps of g-items i and j. Then bitmap of g-itemset $\{i, j\}$ can be easily generated by bitwise ANDing of bitmaps of i and j. To generate the resulting bitmap of extending g-sequence S through I-step or S-step, assume bit K corresponds to event e. Bit K will be the first bit with value 1 in the resulting bitmap, if event e contains the last itemset of generated g-sequence, and all other itemsets are contained in events before e. So, in

order to generate bitmap of a new g-sequence in S-step, suppose the index of the first bit with value one in bitmap B of g-sequence S is b. For each partition in bitmap B, we first set all the bits after bit b to one and all the bits before bit b as well as bit b to 0. Then the resulting bitmap will be obtained by the ANDing operation of the transformed bitmap and the bitmap of the appended g-item i. In the I-step, we do not need any transformation and we directly perform a bitwise ANDing of bitmaps of g-sequence S and the g-item i. To count the support of a g-sequence we can simply count the number of bitmap partitions that contain at least one bit 1. Figures 4.2 and 4.3 show examples of S-step and I-step processes for g-items $(AdmYr, 2009)$ and $(ZIP, 56103)$ in Table 4.1, respectively. Based on the results of these processes, support of the resulting g-sequence $\langle\{(AdmYr, 2009)\}, \{(ZIP, 56103)\}\rangle$ in S-step is 1 and support of the resulting g-sequence $\langle\{(AdmYr, 2009), (ZIP, 56103)\}\rangle$ in I-step is 2.

We also employ some pruning techniques based on Definition 4.7 and Definition 4.11 to decrease the number of candidate g-sequences. Let X with length of l be the current candidate g-sequence, z be a non-inference channel g-item and Z be a g-sequence of length 1 containing z. Assume we want to generate candidate g-sequences of length $l + 1$ from X. The pruning techniques we employ are as follows:

In a S-step,

- if appending g-sequence Z to g-sequence X generates an inference channel, then none of the non-inference channel g-items which are descendants of z should be appended to X in a S-step. For example, if $X = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}\rangle$, $Z = \langle\{(LOS, [1 : 2))\}\rangle$, and g-sequence $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}, \{LOS, [1 : 2)\}\rangle$ is an inference channel, then it is pointless to generate $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}, \{(LOS, 8)\}\rangle$. Because it can not be part of a minimal inference channel.

- if appending g-sequence Z to g-sequence X generates an inference channel, then z and all its descendants should be added to set $lock_S$ of all candidate g-sequences

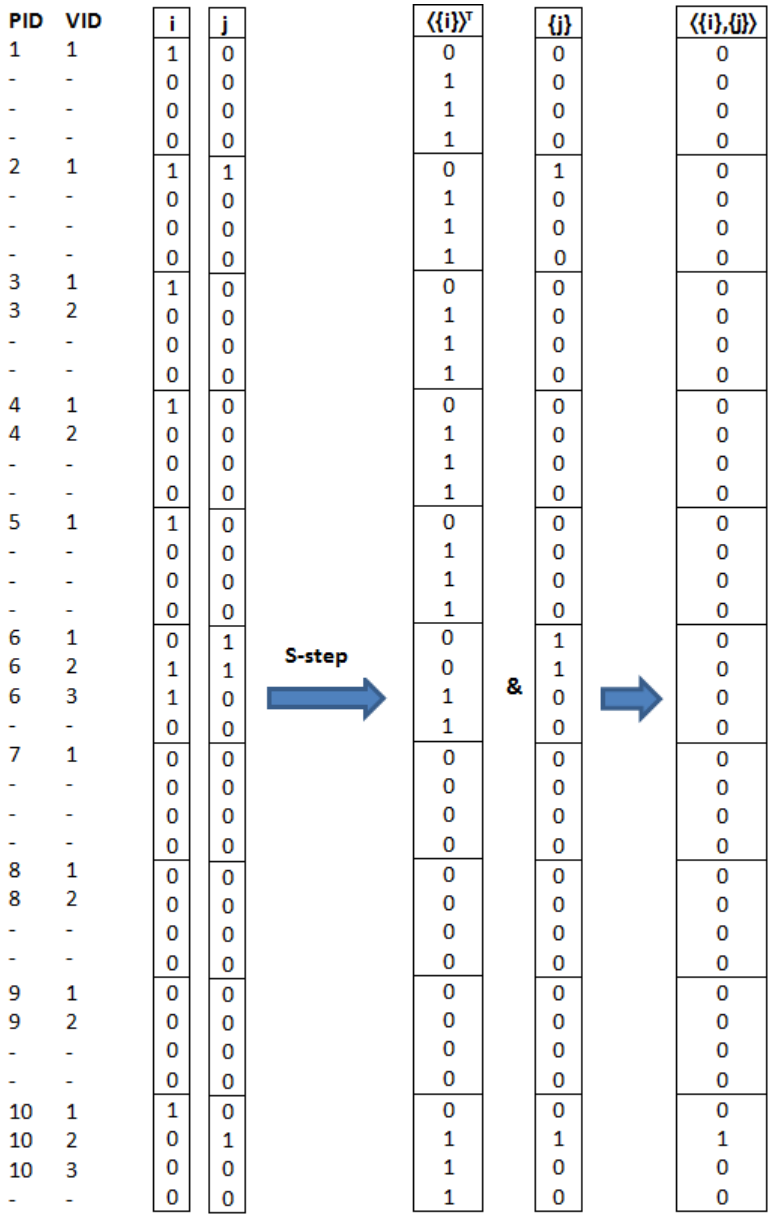


Figure 4.2: S-step processing on the bitmaps of g-items $i = (AdmYr, 2009)$ and $j = (ZIP, 56103)$

X' which are more specific than X, i.e.,$X \leq_S X'$. Let $X = \langle\{(AdmYr, [2005 : 2008]), (LOS, 2)\}, \{(AdmYr, 2010)\}\rangle$ and $Z = \langle\{(LOS, [1 : 2))\}\rangle$. If g-sequence $\langle\{(AdmYr, [2005 : 2008]), (LOS, 2)\}, \{(AdmYr, 2010)\}, \{(LOS, [1 : 2))\}\rangle$ is an inference channel, then we do not need to enumerate, for example, g-sequence $\langle\{(Adm$

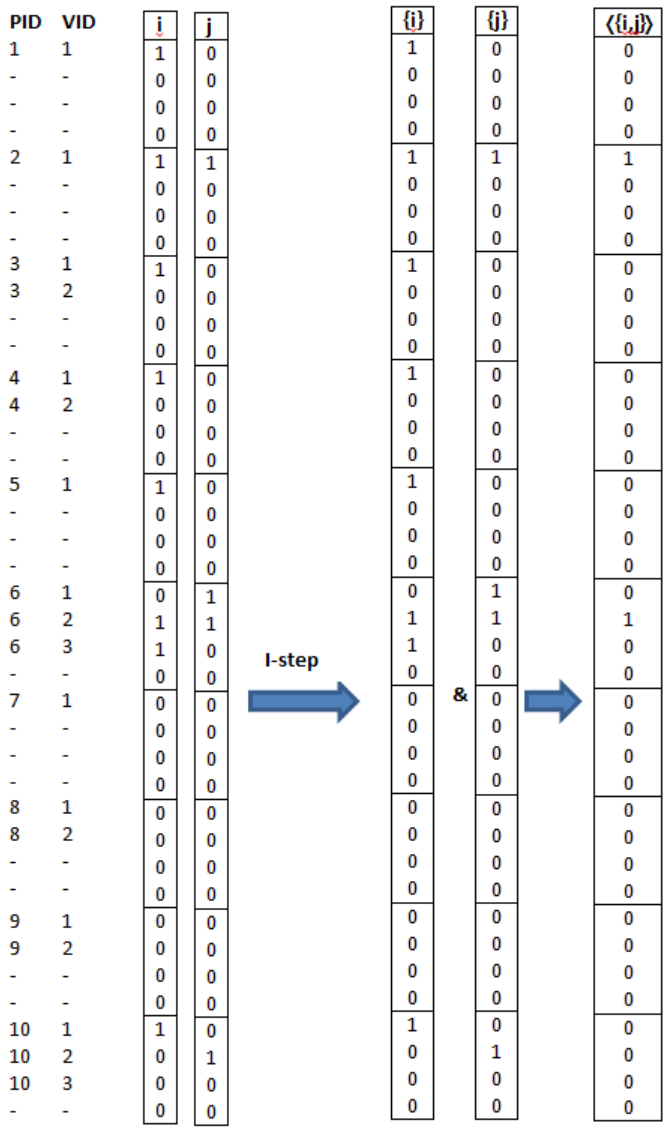


Figure 4.3: I-step processing on the bitmaps of g-items $i = (AdmYr, 2009)$ and $j = (ZIP, 56103)$

$Yr, 2005), (LOS, 2)\}, \{(AdmYr, 2010)\}, \{(LOS, 8)\}\rangle$.

- if appending g-sequence Z to g-sequence X generates an inference channel, then z and all its descendants should be added to sets $lock_S$ and $lock_I$ of all new g-sequences which are generated from X in a S-step. Let $X = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}\rangle$, $Z = \langle\{(LOS, [1:2))\}\rangle$, and $Y = \langle\{(AdmYr, 2009),$

$(LOS, 2)\}, \{(AdmYr, 2010)\}, \{(ZIP, 56107)\}\rangle$ be a g-sequence generated from X in S-step. If g-sequence $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}, \{LOS, [1 : 2)\}\rangle$ is an inference channel, then we do not need to enumerate $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}, \{(ZIP, 56107), (LOS, [1 : 2))\}\rangle$ or $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}, \{(ZIP, 56107)\}, \{(LOS, 8)\}\rangle$.

- if appending g-sequence Z to g-sequence X generates an inference channel, then z and all its descendants should be added to set $lock_S$ of all new g-sequences which are generated from X in a I-step. Let $X = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}\rangle$, $Z = \langle\{(LOS, [1 : 2))\}\rangle$, and $Y = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (ZIP, 56107)\}\rangle$ be a g-sequence generated from X in I-step. If g-sequence $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}, \{LOS, [1 : 2)\}\rangle$ is an inference channel, then we do not need to enumerate $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (ZIP, 56107)\}, \{(LOS, [1 : 2))\}\rangle$ or $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (ZIP, 56107)\}, \{(LOS, 9)\}\rangle$

In an I-step,

- if appending the g-item z to g-sequence X generates an inference channel, then none of the non-inference channel g-items which are descendants of z should be appended to X in an I-step. Let $X = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}\rangle$ and $Z = \langle\{(LOS, [1 : 2))\}\rangle$. If g-sequence $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (LOS, [1 : 2))\}\rangle$ is an inference channel, then we do not need to enumerate $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (LOS, 8)\}\rangle$.

- if appending the g-item z to g-sequence X generates an inference channel, then z and all its descendants should be added to set $lock_I$ of all candidate g-sequences X' which are more specific than X, i.e., $X \leq_S X'$. Let $X = \langle\{(AdmYr, [2005 : 2008]), (LOS, 2)\}, \{(AdmYr, 2010)\}\rangle$ and $Z = \langle\{(LOS, [1 : 2))\}\rangle$. If g-sequence $\langle\{(AdmYr, [2005 : 2008]), (LOS, 2)\}, \{(AdmYr, 2010), (LOS, [1 : 2))\}\rangle$ is an infer-

ence channel, then we do not need to enumerate, for example, g-sequence $\langle\{(AdmYr, 2005), (LOS, 2)\}, \{(AdmYr, 2010), (LOS, 8)\}\rangle$.

- if appending the g-item z to g-sequence X generates an inference channel, then z and all its descendants should be added to set $lock_I$ of all new g-sequences generated from X in a I-step. Let $X = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010)\}\rangle$, $Z = \langle\{(DSFS, [0:1)\}\rangle$, and $Y = \langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (ZIP, 56107)\}\rangle$ be a g-sequence generated from X in I-step. If g-sequence $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (DSFS, [0:1))\}\rangle$ is an inference channel, then we do not need to enumerate $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (ZIP, 56107), (DSFS, [0:1))\}\rangle$ or $\langle\{(AdmYr, 2009), (LOS, 2)\}, \{(AdmYr, 2010), (ZIP, 56107), (DSFS, 3)\}\rangle$

Algorithm 4.1 presents our method of extracting all minimal inference channels.

Algorithm 4.1 Extract Minimal Inference Channels

Input: data D, anonymity threshold K, confidence threshold c, power p, and the set $Sens$ of highly sensitive values

Output: set of minimal inference channels Ω

1: $\Gamma \leftarrow \emptyset$ ▷ set of $NICs$ ordered lexicographically
2: $\Omega \leftarrow \emptyset$
3: $Z \leftarrow$ all g-items sorted in ascending order of their ids
4: **for** $i \in Z$ **do**
5: **if** $sup(i) < K$ or $conf(i \rightarrow s) > c$ **then**
6: $\Omega \leftarrow \Omega \cup \langle\{i\}\rangle$
7: $prune()$
8: **else**
9: $\Gamma \leftarrow \Gamma \cup \langle\{i\}\rangle$
10: $\langle\{i\}\rangle.lock_S \leftarrow$ all ancestors and descendants of g-item i
11: $\langle\{i\}\rangle.lock_I \leftarrow$ all ancestors and descendants of g-item i
12: **end if**
13: **end for**
14: **if** $p > 1$ **then** ▷ continued...

```
15:     while Γ is not empty do
16:         let X ∈ Γ be the g-sequence with the highest order
17:         Γ ← Γ − X
18:         let J_X be the set of all g-sequences of len(X) that have the same prefix as X
19:         for Y ∈ J_X do                                              ▷ S-step
20:             X_S ← generateSExtension(X, Y, X.lock_S, X.lock_I)
21:             if sup(X_S) < K or conf(X_S → s) > c then               ▷ ∀s ∈ Sens
22:                 Ω ← Ω ∪ X_S
23:                 prune()
24:             else
25:                 if len(X_S) < P then
26:                     Γ ← Γ ∪ X_S
27:                 end if
28:             end if
29:         end for
30:         for Y ∈ J_X do                                              ▷ I-step
31:             X_I ← generateIExtension(X, Y, X.lock_S, Y.lock_I)
32:             if sup(X_I) < K or conf(X_I → s) > c then
33:                 Ω ← Ω ∪ X_I
34:                 prune()
35:             else
36:                 if len(X_I) < P then
37:                     Γ ← Γ ∪ X_I
38:                 end if
39:             end if
40:         end for
41:     end while
42: end if
43: return Ω
```

4.3.2 Step 2: Eliminating Inference Channels

The second step is to eliminate extracted inference channels. We propose a greedy algorithm that employs global generalization and global suppression to eliminate all identified $MICs$ to satisfy $(K, C)^P$-privacy. For this purpose, we build a *multi-domain generalization lattice* by combining the generalization hierarchies of individual quasi-identifier attributes. Let $\{H_1, \ldots, H_u\}$ be the set of generalization hierarchies of quasi-identifiers $\{q_1, \ldots, q_u\}$. Every node in the multi-domain generalization lattice corresponds to a vector of u quasi-identifiers' domains with respect to generalization hierarchies $\{H_1, \ldots, H_u\}$. More precisely, every node N is a vector $N = [g_1, \ldots, g_u]$ such that each g_i specifies a generalization level l between 0 and $height(H_i)$ for values of quasi-identifier q_i, where $height(H_i)$ is the height of generalization hierarchy H_i. $l = 0$ corresponds to leaf values (i.e., no generalization) and $l = height(H_i)$ corresponds to the root value of H_i.

A *direct multi-domain generalization relationship* between two vectors of u quasi-identifiers' domains is defined as follows.

Definition 4.12. (Direct Multi-domain Generalization Relationship). Given two vectors $N = [g_1, \ldots, g_u]$ and $N' = [g'_1, \ldots, g'_u]$ of u quasi-identifiers' domains, vector N is a direct multi-domain generalization of vector N', denoted by $<_D$, if

- There is one single value $i \in [1 : u]$ such that $g'_i = g_i - 1$. In other words, values of quasi-identifier q_i with respect to g_i are immediate ancestors of values of q_i with respect to g'_i.
- for all other $j \in [1 : u]$, $i \neq j$, $g_j = g'_j$.

Having nodes $N = [g_1, \ldots, g_u]$, $N' = [g'_1, \ldots, g'_u]$, and $N'' = [g''_1, \ldots, g''_u]$, if $N <_D N'$ and $N' <_D N''$, then node N is an *indirect multi-domain generalization* of N'', denoted by $<_{IN}$. Node N is said to be an ancestor of nodes N' and N'' and nodes N' and N'' are descendants of node N. For instance, considering the dataset in Table 4.1 and generalization hierarchies in Figure 4.1, part of the corresponding multi-domain generalization

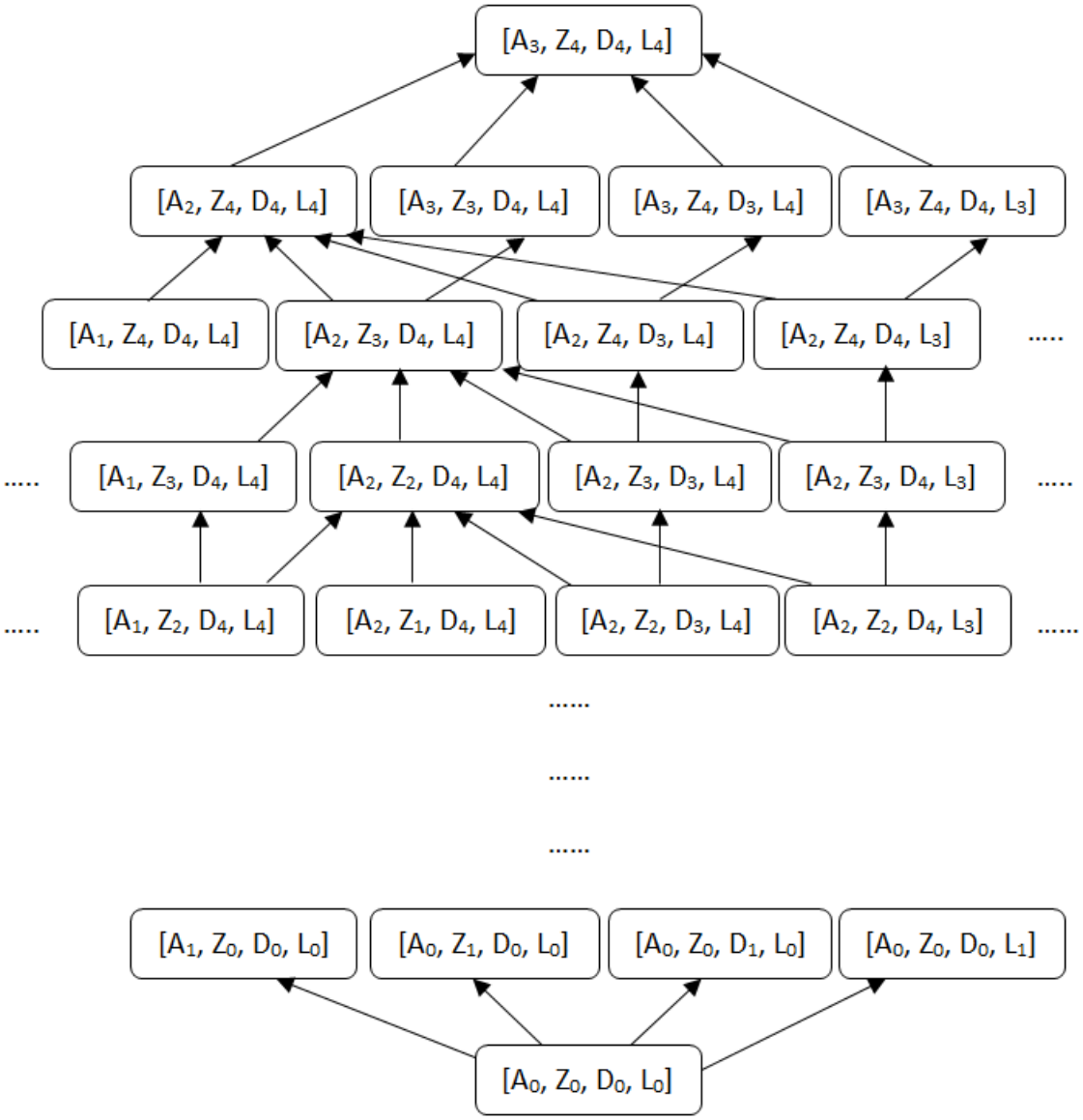


Figure 4.4: multi-domain generalization lattice

lattice is shown in Figure 4.4. For example, node $N = [A_2, Z_3, D_4, L_4]$ means that all values of attribute $AdmYr$ (denoted by A) should be generalized to their corresponding ancestors at level 2 of generalization hierarchy of $AdmYr$, all values of attribute ZIP (denoted by Z) should be generalized to their corresponding ancestors at level 3 of generalization hierarchy of attribute ZIP, and all values of attributes $DSFS$ (denoted by D) and LOS (denoted by L) should be generalized to their corresponding ancestors at

level 4 of generalization hierarchies of $DSFS$ and LOS, respectively. Node N will be a direct multi-domain generalization of nodes $N_1 = [A_1, Z_3, D_4, L_4]$, $N_2 = [A_2, Z_2, D_4, L_4]$, $N_3 = [A_2, Z_3, D_3, L_4]$, and $N_4 = [A_2, Z_3, D_4, L_3]$ and an indirect multi-domain generalization of nodes $N'_1 = [A_1, Z_2, D_4, L_4]$, $N'_2 = [A_2, Z_1, D_4, L_4]$, $N'_3 = [A_2, Z_2, D_3, L_4]$, $N'_4 = [A_2, Z_2, D_4, L_3]$. A multi-domain generalization lattice over u single-domain generalization hierarchies $H_j, j \in [1, u]$ with respect to direct multi-domain generalization relation $<_D$ is a complete lattice of vectors of u domains [57].

We search the multi-domain generalization lattice in a top-down, breadth-first fashion. We start with the root of the generalization lattice which corresponds to the maximum generalization for all quasi-identifiers. At each node N, we first generate all child nodes N', such that $N <_D N'$, by replacing the generalization level $l > 0$ of one of the quasi-identifiers with lower level $l - 1$. Then we check if every child node of this node can eliminate extracted $MICs$. An MIC X would be eliminated by a node N if the generalization level of at least one of the g-items which are present in X is less than its corresponding level in the multi-domain vector N. For example, let $X = \langle\{(LOS, [0:1))\}\rangle$ be an inference channel. According to the generalization hierarchy of LOS, given in Figure 4.1, the generalization level of LOS's value in X is 1. If the generalization level of LOS at node N is 2, then X will be eliminated from the dataset generated based on the generalization at node N.

For the MICs which cannot be eliminated by the generalization strategy in a node, we find a global suppression strategy to efficiently remove them. We identify the minimum number of g-items for suppression which will remove all remaining MICs in a node, while incur the minimum amount of information loss. For this purpose, we define the *suppression weight* of a g-item ϕ for global suppression:

Definition 4.13. (Suppression Weight). Suppression weight of an item ϕ for global suppression, denoted by $suppWeight$, is defined as

$$suppWeight(\phi) = \frac{|MIC_\phi|}{InfoLoss^D_\Phi(\phi)} \tag{4.9}$$

where $|MIC_\phi|$ is the number of minimal inference channels which contain item ϕ and $InfoLoss^D_\Phi(\phi)$ is the total information loss in the whole data set due to suppressing ϕ, as defined in Equation 4.6.

Suppression weight of an item ϕ represents the the number of MICs which contain g-item ϕ and are being eliminated by suppressing ϕ at the cost of $InfoLoss^D_\Phi(\phi)$. A g-item with a higher suppression weight is a better candidate for suppression, since it eliminates more $MICs$ with less information loss. We should emphasize that depending on the QIs' levels in the current generalization node and g-items' levels in extracted $MICs$, $MICs$ which contain g-item ϕ are not only those ones which explicitly contain g-item ϕ, but also the $MICs$ which contain a descendant g-item of ϕ. This should be considered in calculating the suppression weight of every g-item ϕ.

We sort all g-items existing in the set of $MICs$ which cannot be eliminated by a generalization node, in the descending order of their suppression weight. Then we iteratively *i*) add the g-item ϕ with the highest suppression weight to the suppression strategy, *ii*) remove the $MICs$ which contain ϕ from the list of MICs, and *iii*) update suppression weights of g-items in the remaining $MICs$ till no more MIC remains. Since we use global recoding, it is ensured that no new MIC will be generated during the anonymizing process. After finding the best suppression strategy for all child nodes of the current node, we continue with the node that has the minimum total information loss due to generalization and suppression. We stop our search in the generalization lattice when the information loss of all child nodes of the current node is greater than the information loss of the current node. In other words, when the total information loss cannot be decreased anymore, our algorithm terminates. The best anonymization solution will be the generalization strategy corresponding to the current node in the multi-domain generalization lattice possibly combined with a suppression strategy.

We employ some pruning strategies to improve the performance of our algorithm. **Pruning 1.** In our algorithm, for each node we need to evaluate all its child nodes to select the best one as the next node in our search in the lattice of generalization. Eval-

uating each child node can be time consuming. This is because we must check whether all extracted MICs can be eliminated by the corresponding generalization strategy in that node or not. To decrease the computational cost of this part of the algorithm, we propose a pruning strategy that can reduce the number of MICs which should be evaluated in a child node. Our pruning strategy is based on the *monotonicity* property of generalization nodes in the lattice of generalization.

Property 4.1. (Monotonicity of Generalization Nodes). If a generalization node N' can eliminate minimal inference channels X such that $Sup(X) < K$, then all generalization nodes N such that $N <_D N'$ or $N <_{IN} N'$ can also eliminate all such inference channels X.

From Property 4.1 it can be concluded that if a node N in the generalization lattice cannot eliminate an MIC such that $Sup(MIC) < K$, then none of the nodes N' such that $N <_D N'$ or $N <_{IN} N'$ can eliminate that MIC. Therefore, in our search in the lattice of generalization, for every minimal inference channel X with $Sup(X) < K$ that cannot be eliminated by the generalization strategy in node N, we mark all nodes N' which are immediate descendants of N, i.e., $N <_D N'$, as the nodes that cannot eliminate those minimal inference channels. Therefore, in the next steps when we evaluate a child node of N, we can skip those $MICs$ that are already marked.

Pruning 2. If the information loss of a node N due to a generalization strategy is greater than the total information loss of the best node so far, we prune node N and we do not evaluate it. This is due to the fact that node N cannot anonymize data with a less information loss than the current best node.

Algorithm 4.2 shows the pseudocode of our algorithm.

Algorithm 4.2 Eliminate Minimal Inference Channels

Input: set of minimal inference channels Ω, anonymity threshold K

Output: generalization strategy Λ, suppression strategy Φ, total information loss $InfoLoss(D)$

let *current* be the current node
let C be the set of immediate descendant nodes of current node
let R_x be the set of MICs that cannot be eliminated in node x
let W be a table of pairs $< item, suppWeight(item) >$ for all items in a set of MICs
let MIC_x be the set of MICs which contain item x
let Λ_x and Φ_x be the generalization and suppression strategies in node x
let $InfoLoss_x(D)$ be the total information loss incurred by Λ_x and Φ_x
let IL_C be the set of information loss of descendant nodes of the current node
$current \leftarrow [height(H_1), ..., height(H_u)]$ ▷ initialize the current node to the root of generalization lattice corresponding to the highest level of generalization for all quasi-identifiers
$R_{current} \leftarrow \emptyset$
$\Lambda_{current} \leftarrow current$
$\Phi_{current} \leftarrow \emptyset$
$InfoLoss(D) \leftarrow 1$
while true **do**
    $C \leftarrow getImmediateDescedants(current)$ ▷ generate all nodes c such that $current <_D c$
    for $c \in C$ **do**
        $\Lambda_c \leftarrow c$
        $\Phi_c \leftarrow \emptyset$
        if $InfoLoss_c(D) < InfoLoss(D)$ **then** ▷ pruning 2
            for $MIC \in \Omega$ **do** ▷ check if node c eliminates MICs
                if $(supp(MIC) < K)$ AND $(MIC \in R_{current})$ **then** ▷ pruning 1
                    $R_c \leftarrow R_c \cup MIC$
                else if $\neg(eliminate(c, MIC))$ **then** ▷ check if node c can eliminate MIC
                    $R_c \leftarrow R_c \cup MIC$
                end if
            end for
            $W \leftarrow calculateSuppWeights(R_c)$ ▷ calculate suppWeights of items in MICs in set R_c ▷

continued...

28: **while** R_c is not empty **do**
29: $h \leftarrow$ the item with highest suppression weight in W
30: $W \leftarrow W - \{< h, suppWeight(h) >\}$
31: $R_c \leftarrow R_c - MIC_h$ ▷ remove all MICs containing h from R_c
32: **for** item $i \in W$ **do**
33: **if** $MIC_i \cap MIC_h \neq \emptyset$ **then**
34: $W \leftarrow updateSuppWeight(i)$
35: **end if**
36: **end for**
37: $\Phi_c \leftarrow \Phi_c \cup \{h\}$
38: **end while**
39: $IL_C \leftarrow IL_C \cup \{InfoLoss_c(D)\}$
40: **end if**
41: **end for**
42: $c \leftarrow getBestNode(IL_C)$ ▷ get descendant node with minimum information loss
43: **if** $InfoLoss_c(D) \leq InfoLoss(D)$ **then**
44: $InfoLoss(D) \leftarrow InfoLoss_c(D)$
45: $current \leftarrow c$
46: $\Lambda \leftarrow \Lambda_{current}$
47: $\Phi \leftarrow \Phi_{current}$
48: **else**
49: break
50: **end if**
51: **end while**
52: **return** $\{\Lambda, \Phi, InfoLoss(D)\}$

4.3.3 Cost Analysis

Our anonymization algorithm has two steps. In the first step we extract minimal inference channels (Algorithm 4.1) and in the second step we anonymize data to eliminate extracted inference channels (Algorithm 4.2).

In Algorithm 4.1, in steps 3-12, we find all inference channels and non-inference chan-

nels of length 1. The set of non-inference channels Γ will be used to generate all other g-sequences in the next steps. In steps 14-41, we enumerate all g-sequences of length 2 to P (where P is the power of an adversary in $(K, C)^P$-privacy model). Every g-sequence X in Γ, will be extended through S-extension and I-extension procedures. If a generated g-sequence is not pruned, we assign it to the set of minimal inference channels Ω or the set of non-inference channels Γ based on the support and confidence of the g-sequence. As we explained in Section 4.3.1, to efficiently count the support of each candidate g-sequence, we employ a vertical bitmap representation of longitudinal data. The cost of the Algorithm 4.1 is bounded by the maximum number of bitwise AND operations which are performed to calculate the bitmaps of g-sequences generated in S-steps and I-steps. Let N be the number of non-inference channels of length 1. The total number of g-sequences of length at most P is $O(N^P)$. Therefore, Algorithm 4.1 is bounded by $O(N^P)$. To prove this, we find the number of ways to construct a g-sequence of length l, and then arrange items for each case. Our proof is inspired from [135]. We can model this problem as the problem of finding the number of ways to obtain l as a sum of integers. For instance, in Table 4.3, the number of ways to obtain 5 as a sum of integers is shown.

We can interpret each integer as the length of a g-itemset in a g-sequence of length l. The number of ways to assign items to a g-itemset of length i is $\binom{N}{i}$. So, the total number of g-sequences of length l can be calculated by multiplying all possible choices for each case and summing up all cases, i.e., $\sum_{i_1=1}^{l} \binom{N}{i_1} \sum_{i_2=1}^{l-i_1} \binom{N}{i_2} \dots \sum_{i_l=1}^{l-i_1-\dots-i_{l-1}} \binom{N}{i_l}$.

We calculate an upper bound on the number of g-sequences of length l from the above formula. For each position in a g-sequence of length l we can have N g-items and these g-items can be either in the same g-itemset or in different g-itemsets. Therefore, the upper bound for the number of g-sequence of length l is $2^{l-1}N^l$, and the total number of g-sequences of length at most P, i.e., $1, 2, \dots, P$ will be $\sum_{i=1}^{P} 2^{i-1}N^i \leq 2^{P-1} \sum_{i=1}^{P} N^i \approx 2^{P-1}N^P = O(N^P)$.

In the second step of our anonymization framework, we eliminate extracted inference channels using a greedy algorithm that employs global generalization and global

g_1	g_2	g_3	g_4	g_5
1	1	1	1	1
1	1	1	2	
1	1	2	1	
1	2	1	1	
2	1	1	1	
1	1	3		
1	3	1		
3	1	1		
1	4			
4	1			
5				

Table 4.3: Number of ways to get a 5-sequence

suppression. We employ a multi-domain generalization lattice to enumerate all possible generalization strategies. At each node in the lattice, we check if its child nodes eliminate the extracted MICs. For each node that does not eliminate all MICs, we find a suppression strategy. The algorithm terminates when the information loss cannot be decreased any more. The time complexity of Algorithm 4.2 is bounded by $O(|L|*|\Omega|)$ where $|L|$ is the number of nodes in the generalization lattice and $|\Omega|$ is the number of extracted MICs. However, due to pruning strategies and the efficient data structures we employed in our algorithm, the actual time complexity of Algorithm 4.2 is much smaller than $O(|L|*|\Omega|)$.

4.4 Summary

In this chapter we presented a privacy preserving framework to anonymize longitudinal data. We introduced a new privacy model, $(K, C)^P$-privacy, to prevent identity disclosure and attribute disclosure in longitudinal data and proposed a hybrid anonymization algorithm, called $HALT$. Our algorithm employs novel techniques and pruning strategies to effectively anonymize longitudinal data. In the first step, $HALT$ identifies minimal inference channels from data and in the second step it eliminates identified inference channels by utilizing global generalization and global suppression with the goal of preserving as much data utility as possible. Our experiments on synthetic data, reported in Chapter 6, demonstrate the effectiveness of our proposed approach for longitudinal data anonymization.

Chapter 5

Clustering-based Anonymization of Longitudinal Data

5.1 Introduction

Among all kinds of individuals data, sequence data has its own characteristics and importance, and sequence data mining claims many interesting applications in a large number of domains including finance, medicine, business, and security and surveillance [30]. Individual sequence data consist of sequences (ordered lists) of one-dimensional or multidimensional events which are generated by the underlying actors. Customer purchase history, longitudinal medical records and web click-streams are some examples of event sequence data. Sequence data can be used to discover behavior patterns of individuals through their temporal activities. The knowledge about individuals' behaviors is precious for planning, improving quality of services, detecting behavioral changes, and commercial purposes. For instance, analyzing customer purchase histories gives us a better understanding of customers shopping habits and purchase patterns which can be leveraged for targeted marketing, customer retention and business planning. In healthcare, longitudinal medical records of patients can be used to analyze patients' reactions to a new drug, to predict the future behaviors of patients or to support a diagnosis.

However, sequence data often contain sensitive information and may violate privacy of individuals if published. In event sequence data, every event may have a number of attributes that act as *QIs*. Due to temporal correlation among the events of each sequence, privacy attacks on event sequence data are more complex and diverse compared to simpler data types like relational data or transaction data. In other words, in addition to the values of *QIs* within an event, any combination of *QI* values across events along with the temporal information about these values might lead to a privacy breach. As a result, the increasing ability to collect, manage, and release individuals' sequence data is raising privacy concerns. To illustrate potential privacy attacks in releasing event sequence data, consider Example 5.1:

Example 5.1. Consider a hospital that de-identifies and releases the data of Table 5.1 which contains information of multiple visits of patients in the last five years. Every visit is represented with a multidimensional event and the ordered list of these events corresponds to one sequence. Each event has 5 attributes, including admission year, ZIP code, number of days since the first visit in each year, and the length of stay in the hospital, which all act as *QIs*, as well as one sensitive attribute *diagnosis*. An adversary with some background knowledge about visits of a target individual is able to launch two types of privacy attacks: *identity disclosure* and *attribute disclosure*. In the former, if the adversary finds one or a few matches for her background knowledge in the released data, she will be able to uniquely (or nearly uniquely) re-identify the record of the target individual. For instance, if the adversary knows that Bob had a visit in 2009 and he has been living in ZIP code 56230 from 2010, she can uniquely identify Bobs record, #6, and consequently conclude that Bob has HIV. In case of attribute disclosure, the adversary will be able to infer sensitive value(s) of a target individual with high confidence without identifying an individual's record, if all or most of the matching records to her background knowledge have the same sensitive value(s). For example, if the adversary knows that Bob had a visit in 2007 and later in 2011 he was hospitalized for 3 days, then she can conclude that Bob has HIV since both records matching to her knowledge, #8 and #9,

PID	VID	AdmYr	ZIP	DSFC	LOS	Disease
1	1	2009	56117	0	3	Hepatitis
2	1	2007	56103	0	2	Infection
3	1	2008	56942	0	1	Diabetes
3	2	2010	56942	0	30	Infection
4	1	2008	56107	0	2	Diabetes
4	2	2010	56107	0	35	Flu
5	1	2009	56117	0	3	Diabetes
6	1	2009	56103	0	3	Flu
6	2	2009	56103	10	1	Infection
6	3	2010	56230	0	2	HIV
7	1	2008	56072	0	2	Flu
8	1	2007	56361	0	30	Hepatitis
8	2	2011	56107	0	3	HIV
9	1	2007	56230	0	35	Flu
9	2	2011	56107	0	3	HIV
10	1	2009	56072	0	2	Flu
10	2	2009	56103	13	35	Infection
10	3	2010	56043	0	30	Infection

Table 5.1: Inpatient longitudinal data

have HIV in one of their visits.

A common practice for releasing individuals' data for data analysis without violating privacy is *data anonymization*. Data anonymization techniques aim to modify data such that no sensitive information about individuals can be disclosed from published data while data distortion is minimized to ensure usefulness of data for data analysis applications such as data mining, information retrieval, visualization, and query processing. In order to effectively anonymize multidimensional sequence data, to prevent both identity disclosure and attribute disclosure attacks, temporal correlation among the events of each record should be considered in the anonymization process, and it should be guaranteed that no combination of values of *QIs* within an event and across events

of any record leads to a privacy breach. In this chapter, we study privacy threats in publishing longitudinal data and propose a practical privacy model called (K,C)-*privacy* to address the special challenges of anonymizing longitudinal data. This privacy model ensures that every combination of values of *QIs* within an event and across events of any sequence is shared by at least K sequences, and the probability of inferring any high sensitive value is at most c. Table 5.2 shows an anonymized version of the data in Table 5.1 that satisfies (2, 0.5)-privacy using generalization and suppression with respect to generalization hierarchies in Figure 5.1.

We also propose an anonymization approach inspired from *hierarchical agglomerative clustering* [53] to transform the raw data into a version that satisfies (K,C)-privacy

PID	**VID**	**AdmYr**	**ZIP**	**DSFC**	**LOS**	**Disease**
1	1	2009	56117	0	3	Hepatitis
2	1	[2007:2008]	56***	0	2	Infection
3	1	[2007:2008]	56***	0	[0:12)	Diabetes
3	2	[2009:2012]	56***	0	[0:12)	Infection
4	1	[2007:2008]	56***	0	[0:12)	Diabetes
4	2	[2009:2012]	56107	0	[0:12)	Flu
5	1	2009	56117	0	3	Diabetes
6	1	2009	56***	0	[0:1)	Flu
6	2	2009	56103	[1:2)	[0:12)	Infection
6	3	2010	56***	0	[0:12)	HIV
7	1	[2007:2008]	56***	0	2	Flu
8	1	[2007:2008]	56***	0	[0:12)	Hepatitis
8	2	[2009:2012]	56107	0	[0:12)	HIV
9	1	[2007:2008]	56***	0	[0:12)	Flu
9	2	[2009:2012]	56***	0	[0:12)	HIV
10	1	2009	56***	0	[0:1)	Flu
10	2	2009	56103	[1:2)	[0:12)	Infection
10	3	2010	56***	0	[0:12)	Infection

Table 5.2: Anonymized longitudinal data

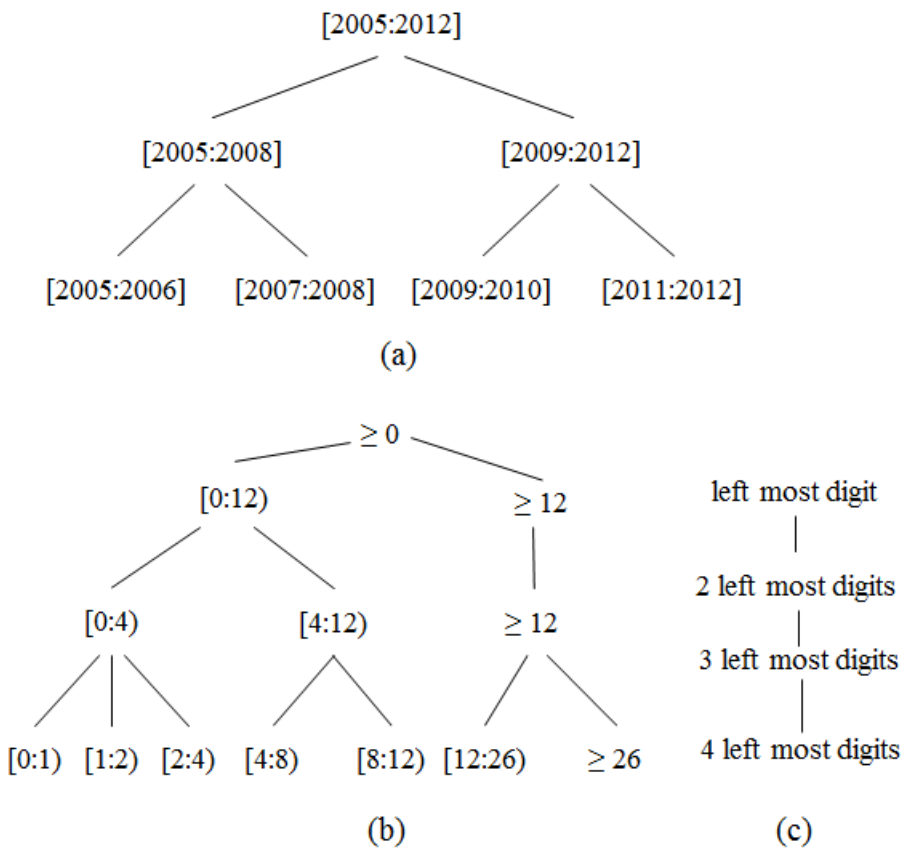


Figure 5.1: Generalization hierarchy for (a) *AdmYr* (b) *DSFC* and *LOS* in terms of number of weeks (c) *ZIP*

but is still useful for data analysis tasks. The key idea underlying our approach is to partition data into clusters such that sequences in a cluster are as similar to each other as possible, and make anonymizations within the clusters. This ensures that less distortion is required when sequences in a cluster are modified to satisfy privacy constraints. A key factor in any clustering algorithm is the distance metric. In order to minimize the overall data distortion due to anonymization, we define the distance between two given sequences as the cost of their optimal anonymization, i.e., information loss. To find the best anonymization of two sequences, we employ techniques from sequence alignment [84] where the goal is to find the best matching between two sequences such that the alignment cost is minimized.

Contributions. We summarize the contributions of this chapter as follows:

1. We propose (K,C)-privacy model to address the challenges of anonymizing longitudinal data

2. We assume an adversary with unbounded knowledge and present an effective algorithm using hierarchical agglomerative clustering and sequence alignment to anonymize longitudinal data with low information loss

3. The results of our experiments confirm that the proposed clustering algorithm can effectively anonymize longitudinal data and retain data utility for data analysis tasks.

5.2 Preliminaries

5.2.1 Sequence Alignment

Finding the best matching between two sequences is known as a *sequence alignment* problem which has found applications in several areas including DNA comparison, natural language processing, and business and marketing research. The basic principle underlying sequence alignment methods is to measure the effort it takes, in terms of specific operations, to make sequences equal. In particular, for two one-dimensional sequences this effort is measured as the smallest sum of cost of deletion, insertion, substitution and identity operations that are needed to align two sequences. One of the most common approaches for sequence alignment is *dynamic programming* [33].

Dynamic programming is an advanced algorithmic technique that solves optimization problems from the bottom up by finding optimal solutions to sub-problems. Dynamic programming solves the sequence alignment problem by breaking it into smaller sub-problems for sub-sequences and then solving each sub-problem. For aligning two one-dimensional sequences $A = a_1, \ldots, a_m$ and $B = b_1, \ldots, b_n$, dynamic programming

creates a two-dimensional score matrix of size $(m+1) \times (n+1)$ with the sequence A along the left side and the sequence B along the top. This score matrix, denoted by S, contains optimal scores of all sub-problems for aligning two sequences.

In general, there are three ways to align two sequences $a_1, \ldots, a_i$ and $b_1, \ldots, b_j$ using optimal alignments of the preceding sub-sequences:

1. optimal alignment of sub-sequences $a_1, a_2, \ldots, a_{i-1}$ and $b_1, b_2, \ldots, b_j$ and deleting a_i

2. optimal alignment of sub-sequences $a_1, a_2, \ldots, a_i$ and $b_1, b_2, \ldots, b_{j-1}$ and inserting b_j into $a_1, a_2, \ldots, a_i$

3. optimal alignment of sub-sequences $a_1, a_2, \ldots, a_{i-1}$ and $b_1, b_2, \ldots, b_{j-1}$, and substituting a_i with b_j

The cost of these three cases can be recursively calculated, using the score matrix S, in terms of the cost of optimal alignments of sub-sequences. The optimal solution of aligning sequences A and B will be the case with the lowest alignment cost.

Let cell $S[i, j]$ be the cost of the optimal alignment of the sequence $a_1, a_2, \ldots, a_i$ and the sequence $b_1, b_2, \ldots, b_j$. The alignment cost of case 1 above is $S[i-1, j]$ plus the cost of deleting a_i. The cost of case 2 is $S[i, j-1]$ plus the cost of inserting b_j, and the cost of the last case is $S[i-1, j-1]$ plus the cost of substituting a_i with b_j. Using this recursive calculations, cell $S[m+1, n+1]$ represents the minimum cost to align two one-dimensional sequences A and B.

Matrix S is filled out progressively from the smallest problem to bigger problems. We start by the left most column and top most row which represent boundary conditions. Let λ and μ be the cost of deletion and insertion operations, respectively. Each cell $S[i, 0]$ in the left most column corresponds to the cost of the optimal alignment of sub-sequence $a_1, a_2, \ldots, a_i$ to "nothing" in sequence $b_1, b_2, \ldots, b_j$. The only possible solution for this sub-problem is deleting a_i. So, $S[i, 0] = \lambda i$. Equivalently, each cell $S[0, j]$ in the top most row corresponds to the cost of the optimal alignment of sub-sequence $b_1, b_2, \ldots, b_j$

to "nothing" in sequence $a_1, a_2, \ldots, a_i$. So $S[0, j] = \mu j$. Obviously $S[0, 0] = 0$. Once we have initialized the left most column and the top most row, we can recursively calculate any other cell in the matrix by using the values of its three adjacent cells on the above, i.e., $S[i-1, j]$, on the left, i.e., $S[i, j-1]$, and on the upper left, i.e., $S[i-1, j-1]$. While we calculate alignment costs, for each cell $S[i, j]$ we keep track of the best operation which is used to get the value of $S[i, j]$, i.e., deletion, insertion, or substitution. We denote these operations with a vertical(V), horizontal(H), and diagonal (D) arrows, respectively, which point back to which of those three adjacent cells used in calculating $S[i, j]$. Once we calculated the values of all cells in the matrix, the cost of optimal alignment is in cell $S[m+1, n+1]$. To construct the resulting aligned sequence, we recursively traceback the arrows from $S[m+1, n+1]$ to $S[0, 0]$ and build the aligned sequence with respect to the corresponding operation at each cell in our path.

5.2.2 Clustering

Clustering is the task of grouping a set of objects such that objects in the same group (called a cluster) are more similar to each other than objects in other groups. Similarity of objects in a clustering problem can be determined in terms of different similarity or distance measures. Clustering methods can be categorized into two groups: hierarchical and non-hierarchical [78]. In hierarchical methods clustering is performed by building a hierarchical tree of clusters. In these methods, the size of a cluster changes monotonically during the clustering process [27]. Hierarchical methods are greedy algorithms which can be performed in two ways:

i. Agglomerative: these are bottom-up algorithms which start by considering each record as a singleton cluster and then successively merge pairs of clusters till all clusters have been merged into a single cluster containing all records.

ii. Divisive: these are top-down algorithms which start by considering all records in one cluster and then recursively split clusters till individual record clusters are reached.

Non-hierarchical clustering methods (aka flat clustering), on the other hand, create a flat set of clusters which do not have any explicit structural relations to each other [78]. Non-hierarchical clustering techniques can be categorized into *i*) the algorithms with fixed number of groups (FNG) in which the number of resulting clusters should be determined in advance and *ii*) the algorithms with variable number of groups (VNG) in which the number of resulting clusters depends on the distances between records and there is no need to a prespecified number of clusters [27]. *C-means clustering* and *Expectation-Maximization* (or EM) algorithms are two examples of non-hierarchical clustering algorithms.

Non-hierarchical clustering methods are efficient and conceptually simple, but they return an unstructured set of clusters and are non-deterministic. Moreover, most non-hierarchical FNG methods require a preset number of resulting clusters as input that may limit the effectiveness of clustering process. Also, the size of clusters in a VNG method changes non-monotonically during the clustering process. More precisely, the size of a cluster may first increase, then decrease, and then increase again during the process. Hierarchical clustering algorithms, on the other hand, return a hierarchy of clusters that is more informative than the unstructured set of clusters returned by non-hierarchical clustering methods [78]. Deterministic behavior, monotonic variation of the size of clusters, and the flexibility of not being required to determine the number of the resulting clusters in advance are other advantages of hierarchical clustering techniques [27].

Some data anonymization tasks such as K-anonymity can be modeled as a clustering problem. The idea is to group records which incur low information loss when anonymized together into the same equivalence class in order to apply minimum generalization and suppression. More precisely, the similarity measure is defined in terms of the amount of information loss incurred by anonymization. The K-anonymity problem can be viewed as a special case of a clustering problem with an extra constraint on the minimum number of K records in every cluster.

There are a number of papers which have proposed clustering-based approaches for the data anonymization problem [16, 29, 46, 58, 64]. In [16], an agglomerative hierarchical clustering approach called *K-member clustering* is proposed for relational data anonymization. The algorithm randomly selects a record r as the center of a new cluster, and then repeatedly selects K - 1 other records which incur the least information loss within the cluster and adds them to the cluster. In [58] authors proposed the *Mondrian algorithm* which is based on divisive hierarchical clustering. The algorithm starts with a single cluster containing all records in the data set, and then keep greedily splitting the clusters into smaller clusters until no more splitting is possible without violating the K-anonymity constraint. Lin and Wei [64] proposed a clustering technique for K-anonymization based on C-means clustering. For a data set D with n records, the algorithm first sorts all records by their QIs values, and then randomly selects $K = \lfloor \frac{n}{C} \rfloor$ records as the center of C clusters. Then, for every record r in D, the algorithm finds the closest cluster to r, adds r to this cluster and updates the center of this cluster. Recently, a new clustering-based anonymization approach for transaction data was presented in [46]. The algorithm constructs a set of clusters of generalized items instead of generalized transactions while the information loss is minimized during the clustering process. Each cluster satisfies a set of user-defined privacy constraints which are specified based on K-anonymity. Clustering-based anonymization has been also tried on trajectory data. Domingo-Ferrer and Trujillo-Rasua proposed a clustering approach to achieve trajectory K-anonymity [29]. The authors presented a new distance measure for trajectories which naturally considers both spatial and temporal aspects of trajectories and also can process those trajectories without time overlap. They employed an agglomerative hierarchical clustering algorithm to anonymize data while minimizing the sum of intra-cluster distances and ensuring that minimum cluster size is K.

As it can be seen, clustering-based anonymization has been proposed for anonymizing different types of data and it has shown very promising results. However, none of these approaches can be directly applied to our problem of anonymizing multidimensional se-

quence data to prevent both identity disclosure and attribute disclosure. In this work, we propose an anonymization algorithm inspired by *hierarchical agglomerative clustering* to anonymize multidimensional sequence data to satisfy (K,C)-privacy while preserving data utility.

5.3 Problem Definition

In this section we present a novel framework to address privacy issues in publishing longitudinal data. This framework forms the basis of our anonymization methodology. Specifically, we describe the privacy model and the utility measure, followed by a formal problem statement.

5.3.1 Privacy Model

Suppose a data holder wants to release its multidimensional sequence data for data analysis. Let $A = \{A_1, A_2, \ldots, A_n\}$ be a set of attributes and $\Delta = \{\Delta_1, \Delta_2, \ldots, \Delta_n\}$ be the corresponding attribute domains. Each A_z is either a categorical or a numerical attribute. Also assume there is one sensitive attribute Ψ with the domain values $\Delta_\Psi = \{s_1, \ldots, s_l\}$. A multidimensional sequence data D is a collection of records of the form (*SID*, S), where *SID* is a unique id for every individual and S is an ordered list of multidimensional events, denoted by $S = \langle e_1, e_2, \ldots, e_m \rangle$. Each event e has the form $(EID, a_1, a_2, \ldots, a_n, s)$ where *EID* is the events id, a_z is a domain value of $A_z, a_z \in \Delta_z$, and s is a value of the sensitive attribute Ψ, $s \in \Delta_\Psi$. Events of every sequence S are ordered with respect to temporal information of one of the attributes $A_z \in \{A_1, A_2, \ldots, A_n\}$. We refer to the value of the z^{th} QI attribute of the j^{th} event of the sequence S_i by e_{ij}(z) and the value of the sensitive attribute Ψ in the j^{th} event of the sequence S_i is denoted by $e_{ij}(\Psi)$. A subset of attributes $\{A_1, A_2, \ldots, A_n\}$ is assumed to be publicly available, so they act as quasi-identifiers, $QIs \subseteq \{A_1, A_2, \ldots, A_n\}$. The values of the sensitive attribute are naturally private.

We assume an adversary who knows that the record of a target individual exists in a released multidimensional sequence dataset. Moreover, the adversary is assumed to have some background knowledge about the sequential events of a target individual, i.e., the values of some *QIs* as well as the order of these values in some of the events of an individual's sequence. Armed with this background knowledge, the adversary seeks to infer some new sensitive information about this individual from published data by finding some records matching to her background knowledge. More precisely, let X be the background knowledge of adversary about a target individual and $S(X)$ be the set of sequences which contain X in the dataset D. X can be any combination of values of *QIs* within an event or across events of a target individual. If the adversary can find such matching for her background knowledge, then she knows that the record of the target individual is in $S(X)$. If the size of this set, denoted by $|S(X)|$, is not "sufficiently" large, then the adversary may be able to identify the target individual's sequence in $S(X)$ and consequently infer his sensitive values. For example, considering the data in Table 5.1, if the adversary knows that Bob had a visit in 2008 and also was hospitalized for 2 days in 2009, then $S(X) = \{\#6\}$. So, the adversary will uniquely identify Bob's record and can infer that Bob has *HIV*. If the size of $S(X)$ is big enough, the adversary cannot identify the record of a target individual with significant probability; however, she may still be able to infer some sensitive information about the individual, if the percentage of sequences in $S(X)$ containing a common sensitive value σ, i.e., $\frac{|S(X) \cap S(\sigma)|}{|S(X)|}$, is high. To illustrate this case, assume that the adversary knows that Bob had a visit in 2007 and in 2011 he was hospitalized for 3 days, then $S(X) = \{\#8, \#9\}$. Since both sequences #8 and #9 have *HIV* in one of their events, the adversary will infer that Bob has *HIV*, without any effort to identify the sequence belonging to Bob.

Since we assume that adversaries' background knowledge about target individuals can be in the form of any combination of *QIs* values, the worst-case scenario would be an adversary who knows the values of all *QIs* in all events of a target individual. Therefore, to protect privacy of individuals against such adversaries when publishing

multidimensional sequence data, the privacy model should ensure that every sequence in the released data is linked to a sufficiently large number of other sequences and the percentage of sequences with the same sensitive value in every group of indistinguishable sequences is not too high. However, the latter case may not need to be satisfied for every value of the sensitive attribute. More precisely, if some values of the sensitive attribute have a lower degree of sensitivity and do not need to be kept private, then we do not need to be worried about these values being too frequent in a group. For example, in the context of publishing medical data, it might be allowed to disclose the value "flu" for the sensitive attribute *disease*. Also, if a sensitive value is too common in a specific context, then disclosing this value may not be an invasion of privacy. For example, when most of the patients visiting a clinic have heart problems, then disclosure of the value "heart disease" may be allowed by the clinic [74]. To effectively handle these cases, we define a set $\Omega \subseteq \Psi$, called *highlysensitive* set, which contains those values of the sensitive attribute Ψ which have a high degree of sensitivity and should be kept private. In the presence of this set, our privacy model must ensure that the frequency of sequences which have at least one of the values in Ω in some of their events is not too high in any group of indistinguishable sequences. This brings us to the following definition.

Definition 5.1. ((K,C)-*privacy*). Given an anonymity threshold $K \geq 2$, and confidence threshold $c \in (0, 1]$, a multidimensional sequence dataset D satisfies (K,C)-privacy if

i. each sequence in D is indistinguishable from at least K-1 other sequences with respect to any combination of *QIs* and

ii. the probability of inferring any *high sensitive* value in any group of indistinguishable sequences is at most C.

The (K,C)-privacy model ensures that, for an adversary with some background knowledge about a target individual, the confidence of adversary to link the individual to a

sequence is at most $\frac{1}{K}$ and the confidence of adversary to link the individual to any high sensitive value is at most C.

5.3.2 Information Loss

In order to enforce (K,C)-privacy, we partition data into groups of size at least K indistinguishable sequences that satisfy diversity constraint on the values of sensitive attribute. This process can be viewed as a constraint-based clustering problem with two constraints: *i*) each cluster contains at least K sequences and *ii*) the frequency of high sensitive values in each cluster is at most C. We employ generalization and suppression on the values of *QIs* to modify data and form clusters. We assume that a generalization hierarchy $H_z$is specified for each attribute $A_z \in QIs$ which defines generalization-specialization relation over the domain of an attribute. Leaf nodes represent all the distinct values in the domain of the attribute and the parent nodes represent more general values. Figure 5.1 shows generalization hierarchies of *QIs* of data in Table 5.1. This anonymization process incurs information loss because some original values of *QIs* in every sequence are either replaced with less specific values or are totally removed. Let D^* be an anonymization of the multidimensional sequence data D. D^* corresponds to a set of clusters $C = C_1, C_2, \ldots, C_r$ which is a clustering of sequences in D. All sequences in a given cluster C_j are anonymized together.

We define the amount of information loss incurred by anonymizing D to D^* as

$$IL(D, D^*) = \frac{1}{|D|} \sum_{j=1}^{r} IL(C_j) \tag{5.1}$$

where $IL(C_j)$ is the information loss of the cluster C_j, which is defined as the sum of information loss of anonymizing every sequence S in C_j:

$$IL(C) = \sum_{i=1}^{|C|} IL(S_i, S_i^*) \tag{5.2}$$

where $|C|$ is the number of sequences in the cluster C, and $IL(S_i, S_i^*)$ is the information loss of anonymizing the sequence S_i to the sequence S_i^*.

Each sequence is anonymized by generalizing or suppressing some of the *QIs* values in some of its events. So, we define information loss of a sequence based on the information loss of its events. First, we use the *Loss Metric* (LM) measure [50] to capture the amount of information loss incurred by generalizing the value a of the attribute A to one of its ancestors $\hat{a}$, with respect to the generalization hierarchy H:

$$IL(a, \hat{a}) = \frac{|\mathcal{L}(\hat{a})| - |\mathcal{L}(a)|}{|\Delta_A|} \tag{5.3}$$

where $\mathcal{L}(x)$ is the number of leaves in the subtree rooted at x.

The information loss of each event e is then defined as

$$IL(e, e^*) = \sum_{n=1}^{|QI|} IL(e(n), e^*(n)) \tag{5.4}$$

where e^* is the ancestor of the event e, $e(n)$ is the value of n^{th} QI of the event e and $e^*(n)$ is its corresponding value in the event e^*.

Hence, the information loss incurred by anonymizing each sequence is as follows:

$$IL(S, S^*) = \sum_{m=1}^{|S|} IL(e_m, e_m^*) \tag{5.5}$$

5.3.3 Problem Statement

As the ultimate goal of data sharing is to allow data mining, we should retain as much information as possible in the released data. In other words, we should ensure that the anonymization cost is minimized. We consider the scenario where the data analysis task is unknown at the time of data publication. So, our goal is to transform a multidimensional sequence data into an anonymized version that satisfies (K,C)-privacy and preserves data utility as much as possible. The problem we tackle is formulated as follows:

Definition 5.2. (multidimensional sequence data anonymization). Given multidimensional sequence data D, a (K,C)-privacy constraint, and a generalization hierarchy for every attribute contained in *QIs*, the problem of multidimensional sequence data anonymization is to construct an anonymized version D^* of D such that D^* satisfies (K,C)-privacy while minimizing the overall information loss as defined in Equation 5.1.

5.4 Anonymization Framework

We propose a bottom-up anonymization algorithm based on hierarchical agglomerative clustering. The general idea is to anonymize data by starting with the trivial clustering that consists of singleton clusters and then keep merging the two closets clusters, using generalization and suppression, until all clusters satisfy privacy constraints based on the (K,C)-privacy model. Every cluster has a representative sequence which is being generated by anonymizing all sequences in the cluster. Two clusters are merged by finding the best anonymization of their representatives. A key factor in any clustering algorithm is the distance measure. The distance between two clusters is calculated based on the result of anonymizing their representatives and the clusters with the smallest distance are chosen to be merged.

In order to minimize the overall data distortion due to anonymization, we define the distance between two given clusters as the change in information loss when we merge the clusters. For this purpose, we used one of the distance functions proposed in [45]:

$$dist(X,Y) = \frac{IL(X \cup Y) - IL(X) - IL(Y)}{\log |X \cup Y|} \tag{5.6}$$

where $IL(X \cup Y)$ is the information loss of the merged cluster, and $IL(X)$ and $IL(Y)$ are information loss of clusters X and Y before merge, respectively.

We can compute the information loss of a merged cluster, i.e $IL(X \cup Y)$, based on the information loss of anonymizing representatives of two clusters X and Y:

Definition 5.3. Let $\ddot{X}$ and $\ddot{Y}$ be the representative sequences of the clusters X and Y and M_{XY} be their best anonymization. Then the information loss of the merged cluster $X \cup Y$ is defined as

$$IL(X \cup Y) = (IL(X) + |X| \cdot IL(\ddot{X}, M_{XY})) + (IL(Y) + |Y| \cdot IL(\ddot{Y}, M_{XY})) \tag{5.7}$$

where $IL(\ddot{X}, M_{XY})$ and $IL(\ddot{Y}, M_{XY})$ are information loss of transforming $\ddot{X}$ and $\ddot{Y}$to M_{XY}.

If we instantiate the value of $IL(X \cup Y)$ from Definition 5.3 in Equation 5.6, we will have:

$$dist(X, Y) = \frac{|X| \cdot IL(\ddot{X}, M_{XY}) + |Y| \cdot IL(\ddot{Y}, M_{XY})}{\log |X \cup Y|} \tag{5.8}$$

We use Equation 5.8 to find the distance between clusters X and Y.

In general, two representative sequences have different numbers of events. So, anonymizing these sequences can be modeled as a *sequence alignment* problem where the goal is to find the best matching between two sequences such that the alignment cost is minimized.

5.4.1 Alignment Algorithm

Inspired by [85], we employ dynamic programming to align representatives of clusters with the goal of minimizing anonymization cost. Dynamic programing is simple to implement and implicitly explores the whole solution space. Therefore, it is able to find an optimal alignment given a particular scoring function. Let multidimensional sequences $\ddot{X} = \{x_1, x_2, \ldots, x_l\}$ and $\ddot{Y} = \{y_1, y_2, \ldots, y_t\}$ be representative sequences of two clusters. The best alignment between these sequences is an anonymization of $\ddot{X}$ and $\ddot{Y}$, using generalization and suppression, which incurs minimum information loss.

In our problem of aligning representative sequences $\ddot{X}$ and $\ddot{Y}$, for every $q \in QI$ we create a score matrix, denoted by S_q, to store the cost of all sub-problems for aligning two one-dimensional sequences resulting from projecting sequences $\ddot{X}$ and $\ddot{Y}$ on q. Therefore, we will have $|QI|$ score matrices where the scores correspond to the information loss of alignment solutions for sub-problems with respect to their corresponding $q \in QI$. The optimal alignment of sequences $\ddot{X}$ and $\ddot{Y}$ is the alignment which incurs the minimum information loss considering all *QIs*. The operations which we use to align (anonymize) two sequences are generalization and suppression. Generalization corresponds to identity and substitution operations which are used in one-dimensional sequence alignment methods. Suppression, on the other hand, corresponds to deletion and insertion operations. Therefore, any diagonal move in a score matrix represents generalization and any horizontal or vertical move represents suppression. If two values of the attribute $q \in QI$

are identical, their generalization is equal to the values themselves; otherwise both values are replaced with their *lowest common ancestor* (LCA).

Definition 5.4. (Lowest common ancestor). Given the attribute A and its corresponding generalization hierarchy H_A, lowest common ancestor of values v and w of the attribute A is the lowest node in the generalization hierarchy H_A that is an ancestor of both v and w.

We have three cases for aligning multidimensional sequences $\ddot{X}$ and $\ddot{Y}$:

1. aligning $\{x_1, x_2, \ldots, x_{l-1}\}$ and $\{y_1, y_2, \ldots, y_{t-1}\}$, and generalizing x_l and y_t, which means replacing every QI value in x_l and its corresponding QI value in y_t with their LCA

2. aligning $\{x_1, x_2, \ldots, x_{l-1}\}$ and $\{y_1, y_2, \ldots, y_t\}$, and suppressing x_l

3. aligning $\{x_1, x_2, \ldots, x_l\}$ and $\{y_1, y_2, \ldots, y_{t-1}\}$, and suppressing y_t

Each of these solutions have an anonymization cost and our objective is to find the best alignment with minimum information loss. The cost of each solution is calculated as the sum of its cost for every $q \in QI$. Let $S_q[i, j]$ be the information loss of the best alignment of the sequence prefix $x_1, x_2, \ldots, x_i$ and the sequence prefix $y_1, y_2, \ldots, y_j$ projected on q. Then $S_q[i, j]$ is calculated with respect to three possible sub-problems denoted above. More precisely,

$$S_q[i,j] = min \begin{cases} S_q[i-1, j-1] + I_q(i,j) & (1) \\ S_q[i-1, j] + I_q(i,j) & (2) \\ S_q[i, j-1] + I_q(i,j) & (3) \end{cases}$$

where $I_q(i, j)$ is a function which returns the information loss of the alignment solution

for values of $q \in QI$ in the events x_i and y_j, given by

$$I_q(i,j) = \begin{cases} IL(x_i(q), lca_{i,j}(q)) + IL(y_j(q), lca_{i,j}(q)) & (1) \\ IL(x_i(q), root(H_q)) & (2) \\ IL(y_j(q), root(H_q)) & (3) \end{cases}$$

where $lca_{i,j}(q)$ is the LCA of values of q in the events x_i and y_j, and $root(H_q)$ is the root value of generalization hierarchy H_q.

Once, we have computed the scores for every cell in score matrices, the information loss of the optimal alignment of sequences $\{x_1, x_2, \ldots, x_l\}$ and $\{y_1, y_2, \ldots, y_t\}$ will be the sum of values of cells $S_q[l+1, t+1]$ for every $q \in QI$.

In order to keep track of the best operations, besides the score matrices, we assume a move matrix M where each cell $M[i,j]$ contains the operation which is chosen to align the sequence prefix $x_1, x_2, \ldots, x_i$ and the sequence prefix $y_1, y_2, \ldots, y_j$, i.e., one of the operations diagonal (D), vertical (V) or horizontal (H). To build the sequence $M_{X,Y}$ which is the result of best alignment of sequences $\ddot{X}$ and $\ddot{Y}$, we do a traceback on matrix M from cell $M[l+1, t+1]$ to cell $M[0,0]$. If cell $M[i,j]$ contains D, we add a new event to $M_{X,Y}$, which contains the LCA of every $q \in QI$ value in x_i and its corresponding QI value in y_j. If cell $M[i,j]$ contains V or H, it means suppression of event x_i or y_j respectively. So we skip those events and do not add any new event to $M_{X,Y}$. Algorithm 5.1 provides an overview of our alignment algorithm.

In steps 1-11 we initialize score matrices S_q for every $q \in QI$ as well as move matrix M. Each cell $S_q[i,0]$ and $S_q[0,j]$ contains the information loss for suppressing every q value in the prefix sequences $x_1, x_2, \ldots, x_i$ and $y_1, y_2, \ldots, y_j$ projected on q. Correspondingly, every cell $M[i,0]$ and $M[0,j]$ contain V and H values for the vertical and horizontal moves corresponding to suppression, respectively.

In steps 12-22, for every value of i and j, we find the cost of the best operation to align prefix sequences $x_1, x_2, \ldots, x_i$ and $y_1, y_2, \ldots, y_j$ projected on q, for $q \in QI$. More precisely, in steps 15-17 we calculate the information loss for generalizing x_i and y_i,

suppressing x_i, and suppressing y_j. Then in steps 19-21, we choose the operation with minimum alignment cost and assign it to $M[i,j]$. The corresponding information loss is stored in $S_q[i,j]$. To build the best matching sequence, in steps 29-45, we traceback on matrix M from cell $M[l+1,t+1]$ to cell $M[0,0]$. We also store the list of suppressed events of X and Y to be used later in anonymization of sequences X and Y.

Algorithm 5.1 Multidimensional Sequence Aligner (MSA)

Input: Sequences $X = \{x_1, x_2, \ldots, x_l\}$ and $Y = \{y_1, y_2, \ldots, y_t\}$

Output: The best alignment $M_{X,Y}$ and the suppression lists $supp_X$ and $supp_Y$

1: **for all** $q \in QIs$ **do** ▷ initialize score matrices of quasi-identifiers as well as the move matrix
2: $S_q[0][0] \leftarrow 0$
3: **for all** $i \in [1:l]$ **do**
4: $S_q[i][0] \leftarrow S_q[i-1][0] + IL(x_i(q), root(H_q))$
5: $M[i][0] \leftarrow V$
6: **end for**
7: **for all** $j \in [1:t]$ **do**
8: $S_q[0][j] \leftarrow S_q[0][j-1] + IL(y_j(q), root(H_q))$
9: $M[0][j] \leftarrow H$
10: **end for**
11: **end for**
12: **for all** $i \in [1:l]$ **do** ▷ calculate alignment costs
13: **for all** $j \in [1:t]$ **do**
14: **for all** $q \in QIs$ **do**
15: $M_q[V] \leftarrow S_q[i-1][j] + IL(x_i(q), root(H_q))$
16: $M_q[H] \leftarrow S_q[i][j-1] + IL(y_j(q), root(H_q))$
17: $M_q[D] \leftarrow S_q[i-1][j-1] + IL(x_i(q), lca_{i,j}(q)) + IL(y_j(q), lca_{i,j}(q))$
18: **end for**
19: $b \leftarrow \text{argmin}_{o \in V,H,D}(\sum_{q \in QIs} M_q(o))$ ▷ select the best operation
20: $M[i][j] \leftarrow b$
21: $S_q[i][j] \leftarrow M_q[b]$
22: **end for**
23: **end for** ▷ Continued...

```
24: i ← l
25: j ← t
26: M_{X,Y} ← ∅
27: supp_X ← ∅
28: supp_Y ← ∅
29: while i ≥ 0 or j ≥ 0 do                ▷ traceback to build the best matching M_{X,Y}
30:     if M[i][j] = V then
31:         supp_X ← supp_X ∪ i
32:         i ← i − 1
33:     else if M[i][j] = H then
34:         supp_Y ← supp_Y ∪ j
35:         j ← j − 1
36:     else if M[i][j] = D then
37:         e ← ∅
38:         for all q ∈ QIs do
39:             e ← e ∪ lca_{i,j}(q)
40:         end for
41:         M_{X,Y} ← M_{X,Y} ∪ e
42:         i ← i − 1
43:         j ← j − 1
44:     end if
45: end while
46: return M_{X,Y}, supp_X, supp_Y
```

5.4.2 Anonymization Algorithm

Our clustering algorithm *clustering-based longitudinal data anonymizer* (*CBLDA*) is based on agglomerative hierarchical clustering. We start with the trivial case of singleton clusters and iteratively merge two closest clusters which are determined by applying our multidimensional sequence alignment algorithm. Once, a cluster satisfies (K,C)-privacy, it will not be merged anymore. The selection of two clusters for merging is guided by *i*) the result of aligning representative sequences of two clusters and *ii*) the diversity of

sensitive attribute values in the resulting cluster. Our algorithm tends to select clusters which comply with the diversity constraint of the (K,C)-privacy model and incur minimum information loss when being merged. When we merge two clusters X and Y with representative sequences $\ddot{X}$ and $\ddot{Y}$, all sequences of these clusters are anonymized with respect to the result of aligning $\ddot{X}$ and $\ddot{Y}$.

This includes suppressing some events in sequences of clusters X and Y, and replacing the remaining events of every sequence with their corresponding generalized event in M_{XY}. However, since we only apply generalization on QI values, the values of sensitive attributes in these events remain unchanged.

Since our goal is to build clusters which satisfy (K,C)-privacy, for every cluster we should check if it contains at least K sequences and if the frequency of sequences which have at least one event with a high sensitive value is not greater than C. When we merge two clusters X and Y, the size of the new cluster is simply the sum of the number of sequences in X and Y. For checking the frequency of sensitive values, we should count the number of sequences which have at least one event with a high sensitive value. In order to efficiently count these sequences, for every cluster we use a data structure, denoted by *HighSensList*, to keep track of these sequences. When we merge two clusters X and Y, the number of sequences with high sensitive value in X or Y may decrease. This can happen when some of the events of sequences in clusters X and Y are suppressed. If the suppressed events in a sequence are the only ones which contain a high sensitive value, then this sequence will not contain any high sensitive value after applying suppression. So it should be removed from the *HighSensList* of the cluster where it is a member. Hence, after applying anonymization on sequences of clusters X and Y, we first update *HighSensLists* of these clusters and then merge two *HighSensLists* to build the *HighSensList* of the new merged cluster. We keep merging clusters until no more than one cluster is left. If the remaining cluster does not satisfy privacy constraints, we remove all sequences contained in this cluster from the data.

Algorithm 5.2 illustrates an overview of our anonymization approach. We define a

distance matrix to keep track of clusters distances. In steps 13-17, we initialize the distance matrix with the distances of all singleton clusters. Method $dist()$ calculates the distance of two clusters based on Equation 5.6. We consider two sets H and G which include clusters that satisfy (K,C)-privacy and does not satisfy (K,C)-privacy, respectively, and initialize set G with all singleton clusters. Then, we iteratively merge the two closest clusters and, at each iteration, if the newly formed cluster satisfies (K,C)-privacy we add it to the set H, otherwise, we add it to the set G of violating clusters which need to be merged further in order to satisfy privacy (steps 18-35). In the latter case, we update the distance matrix to reflect the distance between the new cluster and the original clusters (steps 31-33). We repeat this till no more than one cluster left in G. If this cluster satisfies privacy constraints, we add it to H, otherwise we suppress all sequences in it (steps 36-44). Finally, in steps 45-49, we generate anonymized data by anonymizing each sequence according to the representative of the cluster where the sequence belongs.

Algorithm 5.2 CBLDA(D,K,C)

Input: sequence dataset D, anonymity threshold K, confidence threshold C

Output: anonymized dataset D^* which satisfies (K,C)-privacy

1: let G and H be the sets of clusters not satisfying and satisfying privacy constraints, respectively, and $distMatrix$ be the distance matrix
2: $D^* \leftarrow \emptyset$, $H \leftarrow \emptyset$, $G \leftarrow \emptyset$
3: **for** $s \in D$ **do** ▷ initialize G with singleton clusters containing sequences in D
4: let P be an empty cluster
5: $P = \{s\}$
6: $rep_P \leftarrow s$
7: $G \leftarrow G \cup \{P\}$
8: **end for** ▷ Continued...

9: **for** $P \in G$ **do** ▷ compute distance matrix
10: **for** $T \in G$ **do** ▷ Find the distance of every cluster from all other clusters
11: $distMatrix[ind_P][ind_T] \leftarrow dist(P, T, MSA(rep_P, rep_T))$
12: **end for**
13: **end for**
14: **while** $|G| > 1$ **do**
15: $\{P, T\} \leftarrow \text{argmin}_{X,Y \in G}\, distMatrix[ind_X][ind_Y]$
16: $\{M_{PT}, supp_P, supp_T\} \leftarrow MSA(rep_P, rep_T)$ ▷ resulting sequence and suppression lists of aligning P and T
17: $update(HighSensList_P, supp_P)$
18: $update(HighSensList_T, supp_T)$
19: $HighSensList_P = HighSensList_P \cup HighSensList_T$
20: $P \leftarrow P \cup T$
21: $G \leftarrow G - \{T\}$
22: $rep_P \leftarrow M_{PT}$
23: **if** satisfyPrivacy(P,K,C) **then**
24: $H \leftarrow H \cup \{P\}$
25: $G \leftarrow G - \{P\}$
26: **else**
27: **for** $X \in G - \{P\}$ **do** ▷ update the distance matrix
28: $distMatrix[ind_P][ind_X] \leftarrow dist(P, X, MSA(rep_P, rep_X))$
29: **end for**
30: **end if**
31: **end while**
32: **if** $|G| = 1$ **then**
33: let P be the last cluster in G
34: **if** satisfyPrivacy(P,K,C) **then**
35: $H \leftarrow H \cup \{P\}$
36: $G \leftarrow G - \{P\}$
37: **else**
38: suppress all sequences in P
39: **end if**
40: **end if** ▷ Continued...

41: **for** $P \in H$ **do** ▷ anonymize sequences of every cluster P with respect to rep_p
42: let s be a sequence in P
43: $D^* \leftarrow D^* \cup anonymize(s, rep_p)$
44: $P \leftarrow P - \{s\}$
45: **end for**
46: **return** D^*

5.4.3 Cost Analysis

Our clustering algorithm takes $O(|D|^3)$ time. More specifically, computing distance matrix in steps 13-17 requires $O(|D|^2)$ time. While loop in step 18 takes $O(|D|-1)$, because we have $|D|$ clusters at the beginning and at each iteration two clusters are merged. Finding closest clusters in step 20 corresponds to a linear search of the distance matrix. The cost of such search for the i^{th} iteration is proportional to the current number of clusters squared, i.e., $O((|D|-i-1)^2)$. Finally, steps 29-32 which update the distance matrix after merging two closest clusters, requires $O(|D|-i+1)$ time. Therefore, the time complexity of the algorithm $CBLDA$ is $O(|D|^3)$.

To calculate the distance of two clusters, we need to find the best alignment for representative sequences X and Y of the two clusters. For this purpose, we use our proposed alignment algorithm, MSA, which is based on dynamic programming. The time complexity of MSA is $O(|X||Y|)$. More precisely, steps 1-11 requires $O(|X|+|Y|)$ time. The cost of steps 12-23, which are filling the score matrix S_q for every $q \in QI$ as well as the move matrix M, is $O(|X||Y|)$. Finally, the traceback step (i.e., steps 30-46) to build the best matching sequence for sequences X and Y takes $O(|X|)$ time, assuming $|X| \geq |Y|$. So the total time complexity of MSA algorithm is

$$O(|X|+|Y|) + O(|X||Y|) + O(|X|) = O(|X||Y|)$$

PID	VID	AdmYr	LOS	Diagnosis
1	1	2009	3	Hepatitis
2	1	2010	2	Infection
3	1	2008	1	Diabetes
3	2	2010	6	Infection
4	1	2009	3	Flu
4	2	2009	1	Infection
4	3	2010	2	ulcer
5	1	2007	4	Hepatitis
5	2	2011	3	HIV

Table 5.3: Original Data

5.4.4 Running Example

In this section, we illustrate our anonymization algorithm. Given $K = 2$, $C = 0.4$, and $HighSensValues = \{HIV\}$, our goal is to anonymize the longitudinal data D in Table 5.3 to satisfy (2,0.4)-privacy, using algorithm $CBLDA$ (i.e., Algorithm 5.2). Dataset D has two quasi-identifiers $AdmYr$ and LOS as well as one sensitive attribute *diagnosis*. At the first step, we create singleton clusters containing sequences in D. Also, we definr sets H and G (steps 5-6), and initialize G with singleton clusters. So we will have $H = \{\}$ and $G = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$. At this stage representative sequence of every cluster is equal to the sequence which is in the cluster. Then in steps 13-17, we compute the distance matrix for all clusters in G using our sequence alignment algorithm MSA. The distance between clusters X and Y is calculated based on Equation 5.6.

For example, to compute the distance between clusters {3} and {4}, we first find a matching between representative sequences of these clusters. Assume that we represent the sequence of every individual in table 5.3 as an ordered list of events of the form ($AdmYr$, LOS, $Diagnosis$). Therefore, representative sequences of clusters {3}

	Ø	2009	2009	2010
Ø	0	1	2	3
2008	1	2	3	4
2010	2	1.5	2.5	3

(a) $AdmYr$

	Ø	3	1	2
Ø	0	1	2	3
1	1	0.006	1	2
6	2	1.006	0.012	1.006

(b) LOS

Figure 5.2: Score matrices of quasi-identifiers in table 5.3

and $\{4\}$ will be represented as $rep_{\{3\}} = \langle(2008, 1, Diabetes)(2010, 6, Infection)\rangle$ and $rep_{\{4\}} = \langle(2009, 3, Flu)(2009, 1, Infection)(2010, 2, ulcer)\rangle$, respectively. In steps 1-11 in algorithm MSA (5.1), for every quasi-identifier q we create a score matrix S_q and we initialize first column and first row of each matrix. Also, we define the move matrix M which keeps track of the best operation among {D: diagonal, H: Horizontal, V: vertical} which results in the minimum alignment cost in every cell. More precisely, we create score matrices S_{AdmYr} and S_{LOS} as well as the move matrix M, all of size $(|rep_{\{3\}}|+1)$-by-$(|rep_{\{4\}}|+1)$, i.e., 3-by-4, and initialize $S_{AdmYr}[0][0] = S_{LOS}[0][0] = 0$. Then, at first, we compute the values for every cell $S_{AdmYr}[i][0]$, $S_{AdmYr}[0][j]$, $S_{LOS}[i][0]$, and $S_{LOS}[0][j]$, where $1 \leq i \leq 3$ and $1 \leq j \leq 4$. As we mentioned in Section 5.4.1, every cell $S_{AdmYr}[i][0]$ and $S_{LOS}[i][0]$ contain the information loss of suppressing all $AdmYr$ and LOS values in events 1 to i in $rep_{\{3\}}$, respectively. Equivalently, every cell $S_{AdmYr}[0][j]$ and $S_{LOS}[0][j]$ contain the information loss of suppressing all $AdmYr$ and LOS values in events 1 to j in $rep_{\{4\}}$, respectively. For instance, $S_{LOS}[0][2]$ represents the information loss of suppressing LOS values in the first and second events of sequence $rep_{\{4\}}$ which is equal to $S_{LOS}[0][1] + 1$.

In the next step (i.e., steps 12-23) we start dynamic programming and compute the values of remaining cells in the score matrices and fill out the move matrix accordingly. The resulting score matrices of quasi-identifiers $AdmYr$ and LOS as well as the move matrix M are shown in Figures 5.2a, 5.2b, and 5.3.

For example, to calculate the alignment cost in cell $S_{AdmYr}[2][1]$, we consider the following three possible solutions corresponding to diagonal, vertical, and horizontal moves,

	Ø	(2009,3)	(2009,1)	(2010,2)
Ø	←	H	H	H
(2008,1)	V	D	D	D
(2010,6)	V	D	D	D

Figure 5.3: Move matrix

as discussed in Section 5.4.1, using generalization hierarchies in Figure 5.1:

i. $S_{AdmYr}[1][0]+IL(2010, lca(2010, 2009))+IL(2009, lca(2010, 2009)) = 1+\frac{2}{8}+\frac{2}{8} = 1.5$

ii. $S_{AdmYr}[1][1] + IL(2010, root(H_{AdmYr})) = 2 + \frac{8}{8} = 3$

iii. $S_{AdmYr}[2][0] + IL(2009, root(H_{AdmYr})) = 2 + \frac{8}{8} = 3$

where $lca(2010, 2009)$ represents the lowest common ancestor of values 2009 and 2010, in the generalization hierarchy of quasi-identifier $AdmYr$, i.e [2009 : 2010]. We calculate values of $IL(2010, lca(2010, 2009))$ and $IL(2009, lca(2010, 2009))$ based on Equation 5.3.

Among these three values, dynamic programming chooses the solution with minimum alignment cost. Therefore, the best solution is i and its corresponding cost is stored in cell $S_{AdmYr}[2][1]$. Accordingly, we put value D, which stands for a diagonal move (i.e., generalization), in cell $M[2][1]$. Once we calculated the values of all cells in matrices S_{AdmYr} and S_{LOS}, the costs of optimal alignment for $rep_{\{3\}}$ and $rep_{\{4\}}$ projected on $AdmYr$ and LOS are in cells $S_{AdmYr}[|rep_{\{3\}}|+1][|rep_{\{4\}}|+1]$ and $S_{LOS}[|rep_{\{3\}}|+1][|rep_{\{4\}}|+1]$, respectively. The total alignment cost of aligning $rep_{\{3\}}$ and $rep_{\{4\}}$ is the sum of alignment cost for every quasi-identifier. From Figure 5.2, it can be seen that the total alignment cost is 4.006. To build the best matching sequence, in steps 30-46, we traceback on matrix M from cell $M[|rep_{\{3\}}|+1][|rep_{\{4\}}|+1]$ to cell $M[0][0]$. This path is highlighted in Figure 5.3.

The result of aligning $rep_{\{3\}}$ and $rep_{\{4\}}$ is returned to the algorithm $CBLDA$ (step 15) and is used to calculate the distance of clusters $\{3\}$ and $\{4\}$. Using Equation 5.6, distance of these clusters is $\frac{4.006-0-0}{\log(2)} = 13.35$. We calculate the distance of other clusters in the same way. The resulting distance matrix *distMatrix* is shown in Figure 5.4.

	1	2	3	4	5
1	0	1.68	8.35	13	10
2	1.68	0	6.68	13	10.2
3	8.35	6.68	0	13.35	5.04
4	13	13	13.35	0	16.7
5	10	10.2	5.04	16.7	0

Figure 5.4: Distance matrix

After initializing the distance matrix, we iteratively merge the two closest clusters and at each iteration if the newly merged cluster satisfies (2,0.4)-privacy we add it to set H, otherwise we add it to set G for further anonymization (steps 18-45 in $CBLDA$ algorithm). More precisely, consider $distMatrix$ in Figure 5.4. It can be seen that clusters $\{1\}$ and $\{2\}$ are the closest clusters among 5 clusters. So we have $P = \{1\}$ and $T = \{2\}$, $M_{PT} = \langle([2009 : 2010], [0 : 1))\rangle$, $supp_P = \emptyset$, and $supp_T = \emptyset$. Since none of these clusters have a sequence with high sensitive value HIV, we do not need to update their $HighSensList$. Then we merge two clusters and set the representative of the new cluster using M_{PT} in steps 24-26. So we will have $P = \{1, 2\}$, $G = \{\{1, 2\}, \{3\}, \{4\}, \{5\}\}$, and $rep_P = \langle([2009 : 2010], [0 : 1))\rangle$. Since $|P|= 2$ and $\frac{|(S(HIV):S \in P|}{|P|} = 0$, cluster P satisfies (2,0.4)-privacy. So we add P to set H and update set G (steps 27-29).

In the next iteration we have $H = \{\{1, 2\}\}$ and $G = \{\{3\}, \{4\}, \{5\}\}$. From $distMatrix$ in Figure 5.4, we choose the next closest clusters, i.e., clusters $\{3\}$ and $\{5\}$. So we have $P = \{3\}$ and $T = \{5\}$, $M_{PT} = \langle([2007 : 2008], [0 : 1))([2009 : 2012], [0 : 1]))\rangle$, $supp_P = \emptyset$, and $supp_T = \emptyset$. Since sequence 5 has a high sensitive value HIV in one of its events, we need to update the $HighSensList$ of T with respect to $supp_T$. However, $supp_T$ is empty, so we do not need to change the $HighSensList$ of T. Then in steps 23-26 we merge two clusters that results in $P = \{3, 5\}$, $G = \{\{3, 5\}, \{4\}\}$, and $rep_P = \langle([2007 : 2008], [0 : 1))([2009 : 2012], [0 : 1]))\rangle$. In step 27, we need to check if the new cluster satisfies (2,0.4)-privacy. Since $|P|= 2$ it satisfies the anonymity constraint of (2,0.4)-privacy. However, $\frac{|(S(HIV):S \in P|}{|P|} = \frac{1}{2} = 0.5$ which means cluster P does not satisfy the diversity constraint of (2,0.4)-privacy. So we need to merge it fur-

ther. For this purpose, at first we should update the distance matrix to reflect the distance between the new cluster $\{3,5\}$ and cluster $\{4\}$ which is the only other cluster in G. Using the MSA algorithm, the results of aligning representative sequences of clusters $\{3,5\}$ and $\{4\}$, i.e., $rep_{\{3,5\}} = \langle([2007:2008],[0:1))([2009:2012],[0:1]))\rangle$ and $rep_{\{4\}} = \langle(2009,3)(2009,1)(2010,2))\rangle$ are as follows:

- $M_{\{3,5\},\{4\}} = \langle([2005:2012],[0:1))([2009,2012],[0:1))\rangle$
- $supp_{\{3,5\}} = \emptyset$
- $supp_{\{4\}} = [0]$

Therefore, the distance between clusters $\{3,5\}$ and $\{4\}$, based on Equation 5.8 is 10.65. After updating the distance matrix, in the next iteration, set G contains only clusters $\{3,5\}$ and $\{4\}$. Therefore, $P = \{3,5\}$ and $T = \{4\}$. Since $supp_{\{3,5\}} = \emptyset$, there will be no change in the $HighSensList$ of P which contains sequence 5 with high sensitive value HIV. Also, sequence 4 in T does not have any high sensitive value, therefore we do not need to update $HighSensList$ of T. In steps 24-26, we merge P and T. So $P = \{3,5,4\}$ and $G = \{\{3,5,4\}\}$, and $rep_P = \langle([2005:2012],[0:1))([2009,2012],[0:1))\rangle$. In step 27, we should check if P satisfies (2,0.4)-privacy. Since, $|P|= 3$ and $\frac{|S(HIV):S\in P|}{|P|} = \frac{1}{3} = 0.3$, cluster P satisfies (2,0.4)-privacy. Therefore, we add P to set H and remove it from set G. In the next iteration, since G is empty the while loop in step 18 terminates. Finally, in steps 45-49, we anonymize each sequence with respect to the representative sequence of the cluster where the sequence belongs. The anonymized data D* is shown in Table 5.4.

5.5 Summary

In this chapter, we proposed the (K,C)-privacy model to address the challenges of anonymizing longitudinal data. We also proposed an anonymization framework for longitudinal data using sequence alignment techniques and agglomerative hierarchical

PID	VID	AdmYr	LOS	Diagnosis
1	1	[2009:2010]	[0:1)	Hepatitis
2	1	[2009:2010]	[0:1)	Infection
3	1	[2005:2012]	[0:1)	Diabetes
3	2	[2009:2012]	[0:1)	Infection
4	2	[2005:2012]	[0:1)	Infection
4	3	[2009:2012]	[0:1)	ulcer
5	1	[2005:2012]	[0:1)	Hepatitis
5	2	[2009:2012]	[0:1)	HIV

Table 5.4: Anonymized Data

clustering. To the best of our knowledge, this is the first clustering-based approach for longitudinal data anonymization which prevents both identity disclosure and attribute disclosure without making any assumption about the background knowledge of an adversary. Our experimental results, presented in the next chapter, demonstrate the effectiveness of our proposed algorithm for anonymizing longitudinal data with low information loss.

Chapter 6

Experimental Evaluation

6.1 Introduction

In this chapter, we evaluate the performance of our proposed anonymization techniques. One of the biggest challenges in privacy research, particularly in healthcare, has been the lack of publicly available benchmark datasets which are essential in empirical evaluations to compare different algorithms and tools. This thesis has taken an important step in addressing this challenge by generating synthetic data based on two gold-standard health datasets, namely *CMS Inpatient Claims DE-SynPUF* data provided by the *Centers for Medicare and Medicaid Services* (CMS) [1] as well as *Heritage Health Prize*(HHP)[2] claims dataset. In both datasets, each record pertains to a synthetic inpatient claim (hospital visit). Details of these datasets are described in Section 6.2. We evaluated our proposed algorithms in terms of general information loss and the accuracy of answering query workloads on anonymized data which both have been widely used in the privacy literature for evaluating privacy preserving techniques [18, 118, 69, 58, 66, 46, 67, 72]. It is worth noting that in the context of our problem which assumes a one-time data anonymization,

[1] http://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/SynPUFs/DE_Syn_PUF.html

[2] http://www.heritagehealthprize.com/c/hhp/data

empirical evaluation of efficiency of the algorithms is not of high importance. Therefore we do not consider it in our evaluations.

The general purpose information loss metrics capture the effectiveness of anonymization algorithms regardless of the anonymized data application. These metrics are used when the data publisher does not know how the published data will be analyzed by data recipients which is the case most of the time. The information loss of *HALT* method and *CBLDA* method is calculated based on Equations 4.8 and 5.1, respectively.

We also consider the scenario where data recipients are interested in issuing aggregate queries on anonymized data. This is a common task in data analysis applications. For example, a data recipient may issue a count query on an anonymized health data to find the number of patients who were diagnosed with HIV in a specific year. Such queries may not be answered accurately using anonymized data, due to the fact that a generalized value g of the quasi-identifier q in the anonymized data can be interpreted as any leaf value in the subtree rooted at g in the generalization hierarchy H_q. The higher level of generalization of attribute q leads to the higher uncertainty about the actual value of q and decreases the accuracy of answering an aggregate query. Therefore, a good measure to evaluate the performance of anonymization algorithms is the accuracy of answering workloads of queries on generated anonymized data. This approach does not depend on the underlying algorithm and has been widely used to quantify the performance of anonymization algorithms. Consider COUNT() queries in the following format:

```
Q: SELECT COUNT(*)
   FROM D'
   WHERE conditions
```

where *conditions* can be any combination of values of *QIs* in one event or a number of events belonging to a target individual. When data recipients run Q on anonymized data D', the answer of Q should be estimated by computing the probability that an individual's sequence S_i in D' satisfies Q and then summing these probabilities across

all sequences. This means that the answer of evaluating Q over the anonymized data is not accurate and there is an error in the answer. This error, called *Relative Error* (RE) [129], can be defined as the normalized difference between the answer of Q issued on the anonymized data and the answer of Q on the original data. More precisely, $RE(Q) = \frac{|a(Q)-e(Q)|}{a(Q)}$, where $a(Q)$ is the actual answer of Q on the original data and $e(Q)$ is the estimated answer for Q when it is evaluated over the anonymized data. The following example illustrates the computation of an approximate answer for Q.

Example 6.1. Assume that a data recipient has access to the anonymized data D' in Table 5.2 and wants to know the number of patients who were diagnosed with "Diabetes" and hospitalized less than one week, i.e.,

```
Q': SELECT COUNT(*)
   FROM D'
   WHERE Disease = 'Diabetes' AND LOS BETWEEN 0 AND 6
```

The only sequences which may satisfy this query are sequences 3, 4, and 5 since they have LOS = [0:12) and disease = "Diabetes" (Recall that generalized values of LOS are represented in terms of number of weeks). The probability that sequence 3 satisfies this query is $\frac{7}{84}$, because out of 84 days which are leaf values of [0:12) in the generalization hierarchy of the attribute LOS in Figure 5.1, 7 values are between 0 and 6. Sequence 4 has the same probability as sequence 3 in supporting Q and sequence 5 has probability 1 since the value of its LOS is a leaf value which is between 0 and 6. Then the approximate answer for Q using D' is sum of probabilities of sequences 3, 4 and 5, i.e., $2\times\frac{7}{84}+1 = 1.166$. Referring to the original data in Table 5.1, it can be seen that the answer of query Q when it is issued on the original data is 3. Therefore, the relative error for Q is $\frac{|3-1.166|}{3} \approx 0.61$.

To evaluate how well the anonymized data, generated by our anonymization algorithm, supports query answering, we use the *Average Relative Error*(ARE) for a workload of queries [58]. ARE is defined as the mean relative error of all evaluated queries. To calculate the ARE of our anonymization algorithm, we randomly generated a workload

of 1000 COUNT() queries and evaluated them over the anonymized data. Each query is asking for the number of records which contain a random combination of QI values in an event and across events of a target individual record.

We cannot directly compare our proposed methods with others because no method exists that can anonymize longitudinal data to prevent both identity disclosure and attribute disclosure. So we developed some baseline algorithms for comparison purposes. We conducted our experiments on a 1.80 GHz Intel core $i5$ PC with 8 GB RAM. In the next section, we describe the datasets we used in our experiments. Section 6.3 is devoted to experimentation of our clustering-based approach ($CBLDA$) which was introduced in Chapter 5 and in section 6.4 we report the results of our experiments on the $HALT$ method described in Chapter 4.

6.2 Datasets

In our experiments, we worked with two groups of datasets. In this section we describe these datasets in detail and summarize their characteristics.

DE-SynPUFs Data. The CMS Inpatient Claims DE-SynPUF data is provided by Centers for Medicare and Medicaid Services (CMS) for public use. This data contains three years of synthetic claims (2008-2010) for hospital inpatient services. CMS created this synthetic data based on a 5% sample of actual Medicare inpatients claims data from 2008 to 2010. DE-SynPUF data preserves the detailed data structure of key variables in actual claims. Due to file size limitations, CMS has released inpatient data in 20 separate samples. However, all claims of a particular patient from 2008 to 2010 are in the same sample. Each file contained 81 attributes which are summarized in Table 6.1.

We used a subset of these attributes to build data for our experiments. In the following we explain our approach for generating our data. Attributes *ADMTN_ICD9_DGNS_CD*, *CLM_DRG_CD*, *ICD9_DGNS_CD_1 -ICD9_DGNS_CD_10*, *ICD9_PRCDR_CD_1 - ICD9_PRCDR_CD_6*, and *HCPCS_CD_1 - HCPCS_CD_45* contain information about di-

#	Attribute Name	Attribute Description
1	DESYNPUF_ID	Patient Code
2	CLM_ID	Claim ID
3	SEGMENT	Claim Line Segment
4	CLM_FROM_DT	Claims start date
5	CLM_THRU_DT	Claims end date
6	PRVDR_NUM	Provider Institution
7	CLM_PMT_AMT	Claim Payment Amount
8	NCH_PRMRY_PYR_CLM_PD_AMT	NCH Primary Payer Claim Paid Amount
9	AT_PHYSN_NPI	Attending Physician - National Provider Identifier Number
10	OP_PHYSN_NPI	Operating Physician - National Provider Identifier Number
11	OT_PHYSN_NPI	Other Physician - National Provider Identifier Number
12	CLM_ADMSN_DT	Inpatient admission date
13	ADMTN_ICD9_DGNS_CD	Claim Admitting Diagnosis Code
14	CLM_PASS_THRU_PER_DIEM˙AMT	Claim Pass Thru Per Diem Amount
15	NCH_BENE_IP_DDCTBL_AMT	NCH Beneficiary Inpatient Deductible Amount
16	NCH_BENE_PTA_COINSRNC_LBLTY_AM	NCH Beneficiary Part A Coinsurance Liability Amount
17	NCH_BENE_BLOOD_DDCTBL_LBLTY_AM	NCH Beneficiary Blood Deductible Liability Amount
18	CLM_UTLZTN_DAY_CNT	Claim Utilization Day Count
19	NCH_BENE_DSCHRG_DT	Inpatient discharged date
20	CLM_DRG_CD	Claim Diagnosis Related Group Code
21-30	ICD9_DGNS_CD_1 - ICD9_DGNS_CD_10	Claim Diagnosis Code 1 - Claim Diagnosis Code 10
31-36	ICD9_PRCDR_CD_1 - ICD9_PRCDR_CD_6	Claim Procedure Code 1 - Claim Procedure Code 6
37-81	HCPCS_CD_1 - HCPCS_CD_45	Revenue Center HCFA Common Procedure Coding System 1 - Revenue Center HCFA Common Procedure Coding System 45

Table 6.1: CMS DE-SynPUF Data Attributes

agnoses and procedures related to a patient's visit. So, in the context of our work, these attributes are all sensitive as they represent highly stigmatized conditions, and conditions that patients want to conceal due to their privacy concerns, such as *HIV* and abortion [34]. Since our anonymization framework is designed based on one sensitive attribute, we considered one sensitive attribute *diagnosis* in our data. The value of this attribute in every visit of a patient is set based on values of attributes *ADMTN_ICD9_DGNS_CD*, *CLM_DRG_CD*, and *ICD9_DGNS_CD_1 - ICD9_DGNS_CD_10*. If any of these attributes has a highly sensitive value, then we set attribute *diagnosis* to that highly sensitive

Diagnosis	**ICD9 Diagnosis Code**
HIV	042-044
Other pregnancy with abortive outcome	634-639
Ectopic and molar pregnancy	630-633
Physical, psychological, and sexual abuse or assault	995.5, 995.81, 995.82, 995.83, E95, E96, V15.4
Substance abuse or dependence	291, 292, 303, 304, 305
Psychosexual disorders	302
mental retardation	317-319
Plastic surgery for unacceptable cosmetic appearance	V50.1
Aftercare involving the use of plastic surgery	V51

Table 6.2: Highly Sensitive ICD9 Diagnosis Codes

value, otherwise, one of the values of these attributes is randomly chosen and is assigned to attribute *diagnosis*. We considered all *ICD-9*[3] diagnosis codes which indicate *HIV*, abortion, abuse, psychosexual disorders, mental retardation, or plastic surgery as highly sensitive values that should be protected against attribute disclosure [34]. These values are summarized in Table 6.2. The percentage of these highly sensitive values in the *CMS DE-SynPUF* data files is at most 8.51%. This means that the minimum confidence of an adversary to infer a highly sensitive value of a patient in the *CMS DE-SynPUF* data is approximately 0.1.

Next, we explain our approach for specifying quasi-identifiers. The attributes we chose as quasi-identifiers are the attributes that we believe could be employed by an adversary for an identity-disclosure or an attribute-disclosure attack. Values of attributes *CLM_FROM_DT* (start date) and *CLM_THRU_DT* (end date), which indicate claims start date and end date respectively, were identical to values of attributes *CLM_ADMSN_DT* (admission date) and *NCH_BENE_DSCHRG_DT* (discharge date) in approximately 99.8% of the records in 20 sample files. Therefore, we discarded attributes *CLM_FROM_DT* and *CLM_THRU_DT* and only considered attributes *CLM_ADMSN_DT* and *NCH_BENE_DSCHRG_DT* as quasi-identifiers. The other potential quasi-identifiers

[3]http://www.cdc.gov/nchs/icd/icd9cm.htm

D	$\|D\|$	**Max**($\|e\|$)	**Sens%**
CMS-data1	66773	14	8.39
CMS-data2	66494	14	8.06
CMS-data3	66672	15	8.18
CMS-data4	66253	15	8.03
CMS-data5	66414	12	8.23
CMS-data6	66977	15	8.42
CMS-data7	66791	15	8.21
CMS-data8	66490	14	8.20
CMS-data9	66763	12	8.47
CMS-data10	66585	12	8.18
CMS-data11	66425	14	8.51
CMS-data12	66717	15	8.25
CMS-data13	66324	12	8.25
CMS-data14	67024	13	8.14
CMS-data15	66846	12	8.15
CMS-data16	66800	12	8.25
CMS-data17	66495	12	8.13
CMS-data18	66428	13	8.13
CMS-data19	67037	13	8.12
CMS-data20	66514	12	8.27

Table 6.3: Characteristics of CMS datasets

which we included in our data are *CLM_UTLZTN_DAY_CNT*, which indicates the length of stay in the hospital, and *CLM_PMT_AMT*, which indicates a claim payment amount. We excluded all other payment-related attributes *CLM_PASS_THRU_PER_DIEM_AMT*, *NCH_BENE_BLOOD_DDCTBL_LBLTY_AM*, *NCH_BENE_PTA_COINSRNC_LBLTY_AM*, *NCH_BENE_IP_DDCTBL_AMT*, *NCH_PRMRY_PYR_CLM_PD_AMT* as either most of the records had value 0 or the same value for each of these attributes or they cannot act as quasi-identifiers. Inspired from [34], we also defined another quasi-identifier, namely $DSFC$, which indicates the number of days since the first claim computed for each patient for each year. Therefore, in our data we have 5 quasi-identifiers as well as 1 sensitive attribute *diagnosis*. The other attributes in the original *DE-SynPUFs* data are neither sensitive nor quasi-identifier and, hence, they do not have any impact on the

anonymization process. We generated 20 datasets based on 20 *CMS DE-SynPUF* samples files. For each experiment, we executed our algorithm on these 20 datasets and then averaged the results over the runs. Characteristics of 20 datasets are shown in Table 6.3 where $|D|$ indicates the number of records in dataset D, $\max(|e|)$ indicates maximum number of records (i.e., events) belonging to the same patient, and Sens% indicates percentage of high sensitive values in a dataset.

Synthetic Data. We developed a data generator, inspired from *IBM synthetic data generator* [4], to generate synthetic longitudinal data. Our data structure is inspired from the *Heritage Health Prize* (*HHP*) *claims data set*. The *HHP* was a data mining competition to predict the number of days patients will be hospitalized within the next year using historical claims data. The *HHP* data consists of multiple tables such as claims and lab data. For our study, we only focused on claims data which contains information of all claims of patients over a period of three years. The attributes of this data are shown in Table 6.4 [34].

In [34], anonymization of the *HHP* claims data was studied in order to prevent identity disclosure attacks. Authors identified 6 quasi-identifiers for the claims dataset

Attribute Name	**Attribute Description**
MemberID	Unique ID of the patient
ProviderID	Unique ID of the doctor or specialist providing the service
Vendor	Unique ID of the company that issues the bill
PCP	Unique ID of patient's primary care physician
Year	The year of the claim, Y1, Y2, Y3
Specialty	Specialty of care provider
PlaceSvc	Place where the patient was treated
PayDelay	The delay between the claim and the day the claim was paid for
LOS	Length of stay in hospital
DSFS	Days since first service that year
Diagnosis	*ICD-9* code
CPTCode	*CPT* code: Current Procedural Terminology which describes medical, surgical, and diagnostic services

Table 6.4: HHP Data Attributes

[4] http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

Dataset	$\|S\|$	$Avg\|e\|$	$\|QI\|$	**Sens%**	**Avg**($Max\|e\|$)	**Avg(total #events)**
Data_1000_3_2	1000	3	2	36.7	10	3060
Data_1000_3_3	1000	3	3	37	9	3050
Data_1000_3_4	1000	3	4	34.2	9	3062
Data_1000_5_2	1000	5	2	47.2	11	4761
Data_1000_5_3	1000	5	3	45.8	10	4538
Data_1000_5_4	1000	5	4	45	12	4572
Data_1000_10_2	1000	10	2	65.7	17	9466
Data_1000_10_3	1000	10	3	67.1	15	9594
Data_1000_10_4	1000	10	4	68.3	16	9492
Data_10000_3_2	10000	3	2	35.17	10	30964
Data_10000_3_3	10000	3	3	34.61	11	30864
Data_10000_3_4	10000	3	4	34.51	10	30735
Data_10000_5_2	10000	5	2	43.33	11	46228
Data_10000_5_3	10000	5	3	44.86	13	46008
Data_10000_5_4	10000	5	4	48.04	12	46600
Data_10000_10_2	10000	10	2	67.04	18	95016
Data_10000_10_3	10000	10	3	66.96	18	94648
Data_10000_10_4	10000	10	4	66.52	17	95234

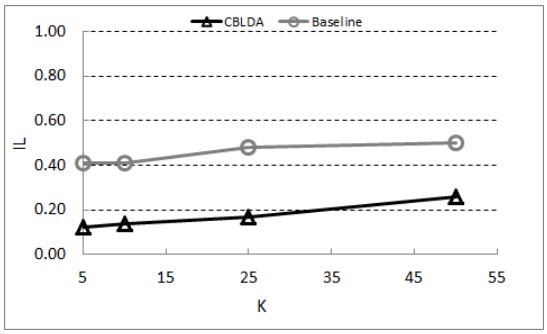
Table 6.5: Synthetic Data Characteristics

including *Specialty*, *PlaceOfService*, *LOS*, *DSFS*, *Diagnosis* and *CPTCode*. Among these attributes we selected attributes *DSFS* and *LOS*. We did not include attributes *Diagnosis* and *CPTCode* since these attributes are sensitive in the framework of our study. Also, we disregarded attributes *PlaceSvc* and *Specialty* as we did not have access to their possible original values in the *HHP* data. Instead, for each claim we considered two other quasi-identifiers, namely *ZIP* code of the patients and *AmdYr* which indicates the year in which a claim took place. The justification for selection of *ZIP* code as a quasi-identifier is that this information is often updated at every visit of a patient and therefore it can be an attribute of a claim [35]. Including *AdmYr* is with respect to attribute *Year* in the *HHP* data. We generated multiple synthetic datasets by varying the number of patients, denoted by $|S|$, which indicates the number of sequences of multi-dimensional events in the longitudinal data, average number of events (i.e., claims) per sequence, denoted by avg $|e|$, and number of *QIs*, denoted by $|QI|$. For every set of data

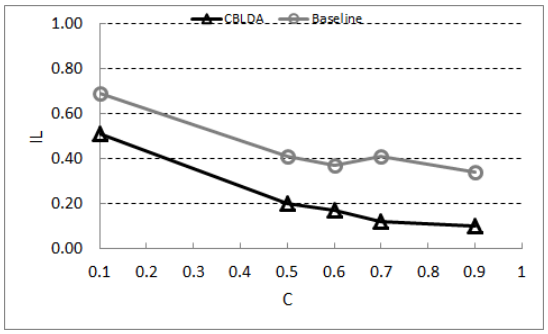
characteristics, we generated 10 datasets. The characteristics of our datasets are summarized in Table 6.5. Sens% indicates the percentage of patients (sequences) with highly sensitive values, Avg($Max|e|$) indicates the average of maximum number of claims per patient, and Avg(total #events) indicates the average of total number of claims in the dataset. We denote each dataset by $Data\_X\_Y\_Z$ where X is the number of sequences (individuals), Y is the average number of events per sequence, and Z is the number of *QIs*. For example, $Data\_10000\_5\_4$ denotes a dataset which has 10000 sequences which on average have 5 events and each event has 4 quasi-identifier. Synthetic datasets with 1000 sequences are denoted by Syn-1000 and datasets with 10000 sequences are denoted by Syn-10000. We executed our proposed algorithms on all datasets generated for every set of data characteristics and reported average of the results over 10 runs as the result for each set of data characteristics.

6.3 Evaluation of CBLDA

In this section we evaluate the effectiveness of CBLDA method on preserving data utility. To demonstrate the benefits of our anonymization technique, we developed a baseline algorithm, called *Baseline*, which is a hierarchical clustering-based approach which does not use dynamic programming for finding the best alignment of clusters' representatives. If two sequences X and Y are of the same size, the baseline algorithm simply applies generalization to every event of two sequences. Otherwise, it first *randomly* suppresses $n = abs(|X|-|Y|)$ events in the longer sequence and then generalizes every remaining events in two sequences. The rational behind designing this baseline algorithm is inspired from machine learning domain where classifiers are compared to a random classifier as a baseline.



(a) IL *vs* K on CMS (C = 0.7)



(b) IL *vs* C on CMS (K = 5)

Figure 6.1: Information loss (IL) of *CBLDA* on *CMS* data

6.3.1 Information Loss

Impact of K. In the first set of experiments, we evaluate the information loss IL of $CBLDA$ method by varying the value of the anonymity threshold K from 5 to 50 while keeping the confidence threshold C fixed, $C = 0.7$. Figure 6.1a shows the information loss for the *CMS* data. As we mentioned in Section 6.2, we executed experiments on 20 *CMS* data files and reported the average of the results. As can be seen, IL increases for both algorithms when K increases. This illustrates the trade-off between privacy and data utility. As K increases, more generalization and suppression is required to ensure that each cluster has at least K indistinguishable records and the probability of inferring sensitive values is at most C. Obviously this leads to more data distortion.

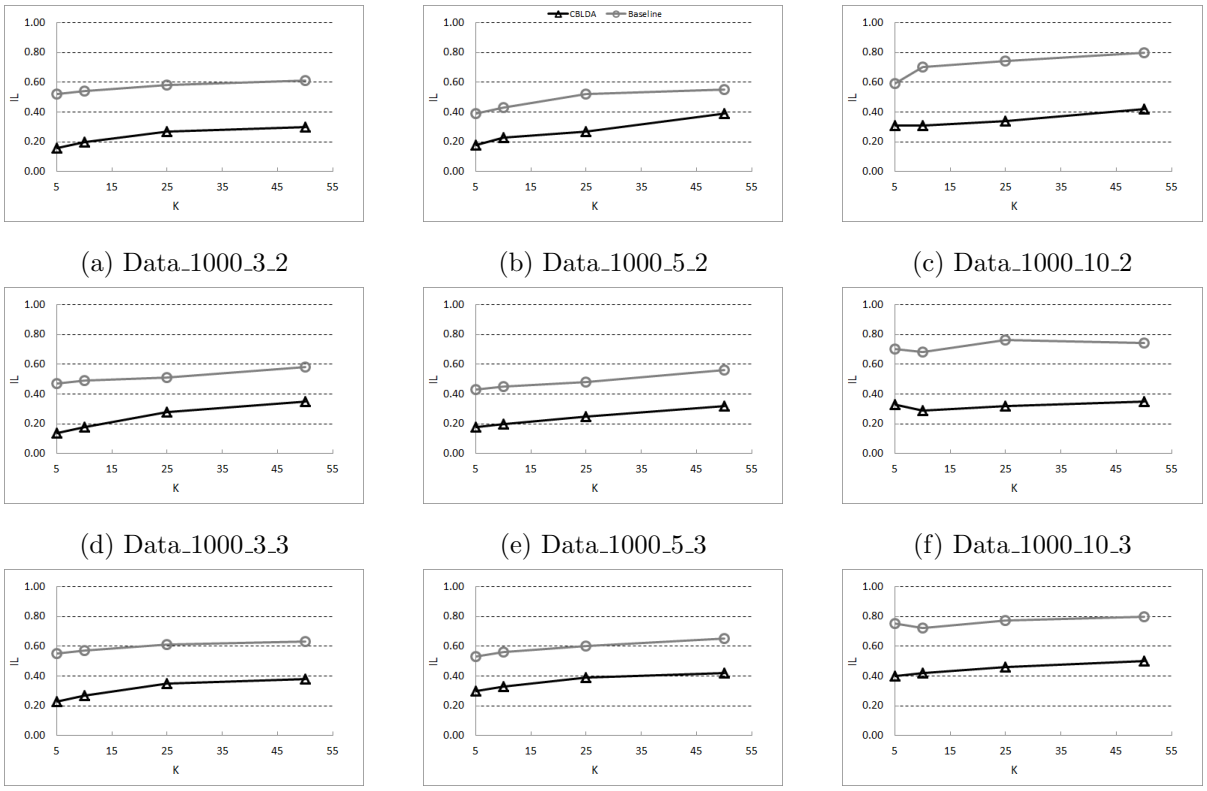
Also, comparing the information loss of our anonymization algorithm based on dynamic programming with the *Baseline* algorithm depicts the benefits of our method. Interestingly, *CBLDA* incurs fairly low information loss (below 0.26) for all values of K between 5 and 50 which is, on average, 62% less than incurred information loss by *Baseline* algorithm. This verifies that employing dynamic programming to align patients' sequences significantly improves the effectiveness of our clustering approach in grouping similar sequences which leads to less generalization and suppression.

The information loss incurred by our algorithm on generated synthetic datasets for various values of K and a fixed confidence threshold is shown in Figures 6.2 and 6.3.

The results show similar trends to *CMS* data and the information loss increases when K increases. In fact, higher values of K require more records to be identical and this leads to more generalization and suppression for larger values of K. Moreover, it can be seen that when the number of QIs increases, information loss increases for any value of K. This is due to the fact that more attributes should be considered in dynamic programming to align two sequences. The same trend can be observed for the average number of events per record in a dataset.

Interestingly, the information loss of $CBLDA$ for all data sets is fairly low. For example, for $K = 5$, which is an acceptable level of anonymity in many applications, the information loss of $CBLDA$ for most data sets in Figure 6.2 is less than 0.3. For larger datasets in Figure 6.3, with an average size 57366 records, $CBLDA$ performs significantly better and the incurred information loss, for $K = 5$, for most data sets is less than 0.2. In particular, $CBLDA$ achieves, on average, 38% improvement on preserving data utility for $K = 5$ for all datasets in Figure 6.3 compared to datasets in Figure 6.2.

Impact of C. In the second set of experiments we change the value of C from 0.1 to 0.9 for a fixed value $K = 5$. This setting allows us to measure the performance of our anonymization algorithm against attribute disclosure for a fixed value of K. The resulting information loss for CMS data and synthetic data is shown in Figures 6.1b, 6.4, and 6.5, respectively. In general, IL decreases when c increases due to a less restrictive privacy requirement. For synthetic data, we can see that the information loss of large data sets with an average number of 10 events per sequence, e.g. $Data\_10000\_10\_3$, for smaller values of C is high. This is due to the percentage of records with a highly sensitive value in these datasets. For example, in $Data\_10000\_10\_3$, 66% of the sequences contain a highly sensitive value, so the information loss is high at C = 0.1, C = 0.5 and C = 0.6. As we increase the value of C, the information loss drops significantly. This is because less generalization and suppression is needed to make clusters which satisfy confidence constraint of (K,C)-privacy model. Overall, larger values of C incur fairly low information loss on all datasets. These results indicate the performance of our algorithm.



(a) Data_1000_3_2 (b) Data_1000_5_2 (c) Data_1000_10_2

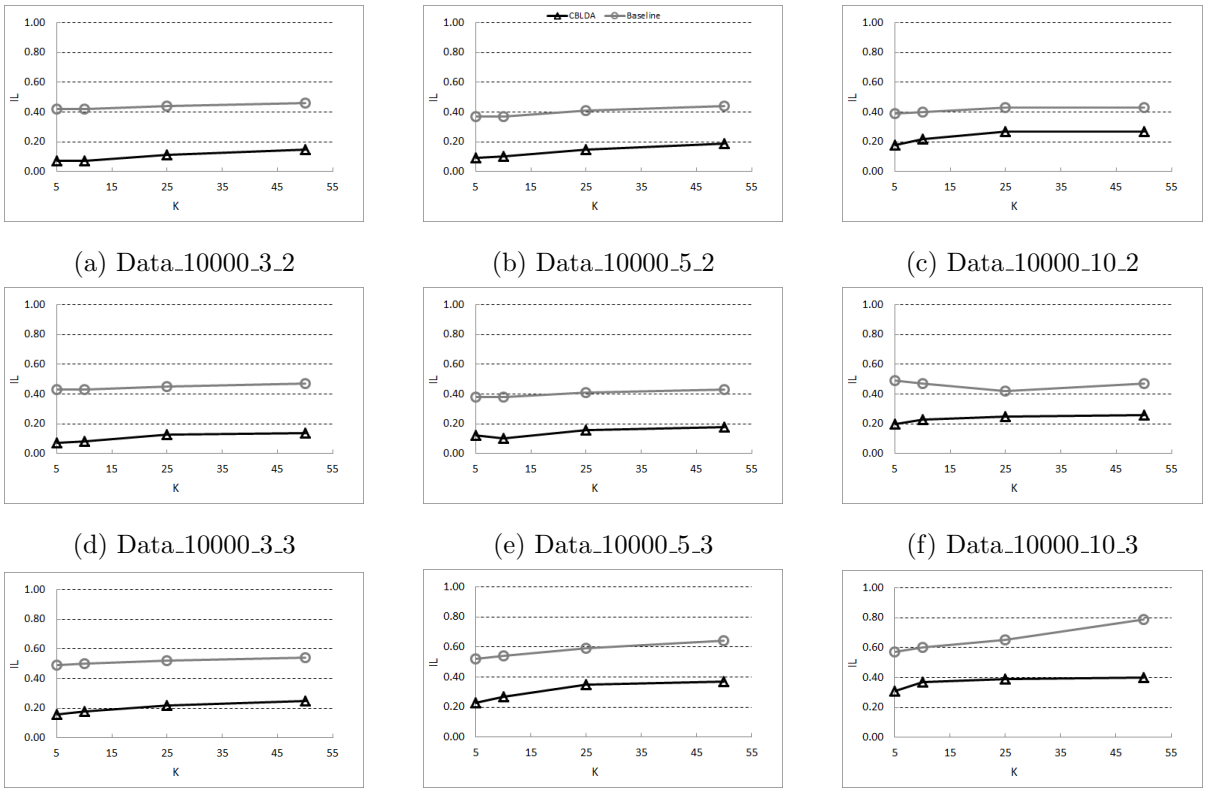
(d) Data_1000_3_3 (e) Data_1000_5_3 (f) Data_1000_10_3

(g) Data_1000_3_4 (h) Data_1000_5_4 (i) Data_1000_10_4

Figure 6.2: Information loss (IL) of *CBLDA* vs k on Syn-1000 (c = 0.7)

6.3.2 Query Answering Accuracy

Impact of K. In the first set of experiments we study how varying the anonymity threshold K affects Average Relative Error (ARE). Figure 6.6a reports resulting ARE of running $CBLDA$ and $Baseline$ algorithms on CMS data, for K values between 5 and 50 while fixing $C = 0.7$. As expected, ARE increases when K increases because higher distortion incurs to satisfy privacy. More precisely, for higher values of K, more generalization and suppression is required to ensure each cluster has at least K records. As a result, the uncertainty in estimating the answer of a COUNT() query issued on anonymized data increases since the probability that a generalized sequence may satisfy



(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3

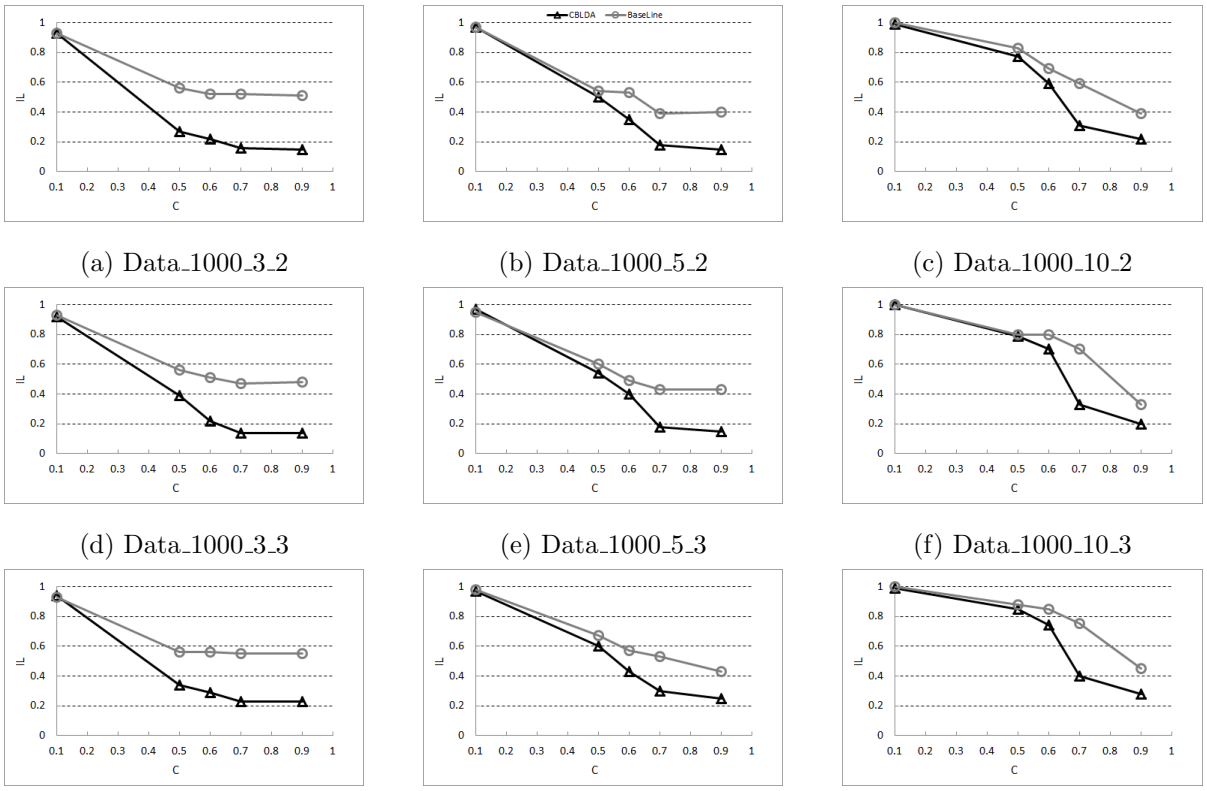
(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

Figure 6.3: Information loss (IL) of *CBLDA* vs K on Syn-10000 (C = 0.7)

the query decreases. Similar results were observed for synthetic datasets which are shown in Figures 6.7 and 6.8.

It is worth noting that $CBLDA$ incurs fairly low ARE for all datasets. In CMS data, value of ARE for all K is at most 1. For Syn-1000 datasets and Syn-10000 datasets, $CBLDA$ has ARE values of at most 1.5 and 1 for most datasets, respectively. Particularly, for $K = 50$, the average value of ARE for Syn-1000 datasets and Syn-10000 datasets is 1.42 and 0.94, respectively. These results imply the effectiveness of $CBLDA$ in answering aggregate queries accurately even for a high level of K.

Impact of C. We also examined the impact of varying the confidence threshold c between 0.1 and 0.9 while fixing $K = 5$. The result of $CBLDA$ and *Baseline* algorithm for CMS data, shown in Figure 6.6b, suggests that ARE is fairly insensitive to



(a) Data_1000_3_2 (b) Data_1000_5_2 (c) Data_1000_10_2

(d) Data_1000_3_3 (e) Data_1000_5_3 (f) Data_1000_10_3

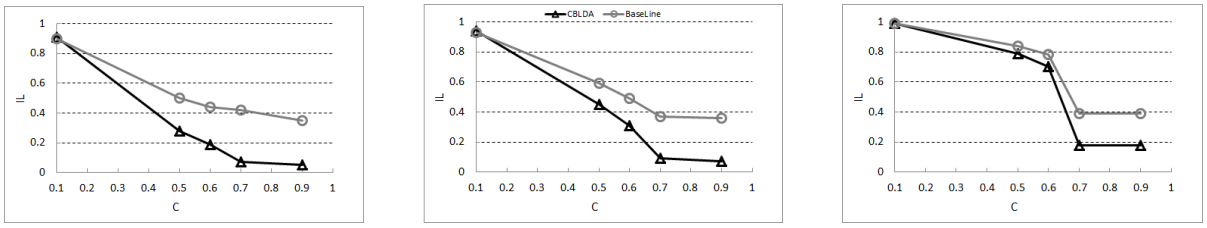
(g) Data_1000_3_4 (h) Data_1000_5_4 (i) Data_1000_10_4

Figure 6.4: Information loss (IL) of *CBLDA* vs C on Syn-1000 (K = 5)

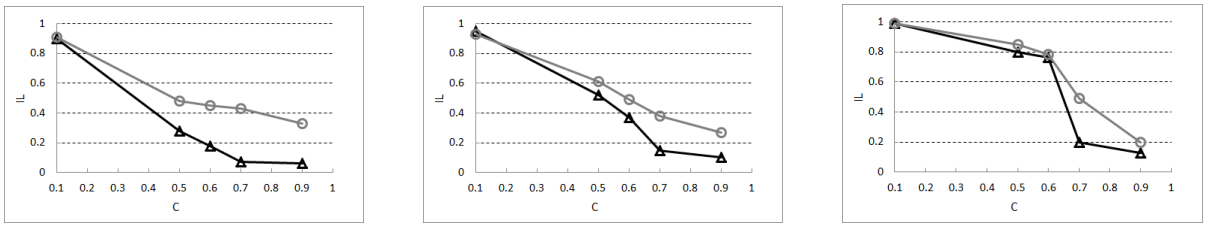
the change of C, when $C \geq 0.5$. This is because approximately 10% of the sequences in CMS data contain a highly sensitive value, so the ARE is high at $C = 0.1$. When c increases, ARE becomes invariant of C. Note also that *Baseline* has higher ARE than $CBLDA$ in both cases. This again confirms the utility enhancement due to the employment of dynamic programming in $CBLDA$. Figures 6.9 and 6.10 report the results for synthetic data. The same results has been observed for synthetic datasets.

6.4 Evaluation of HALT

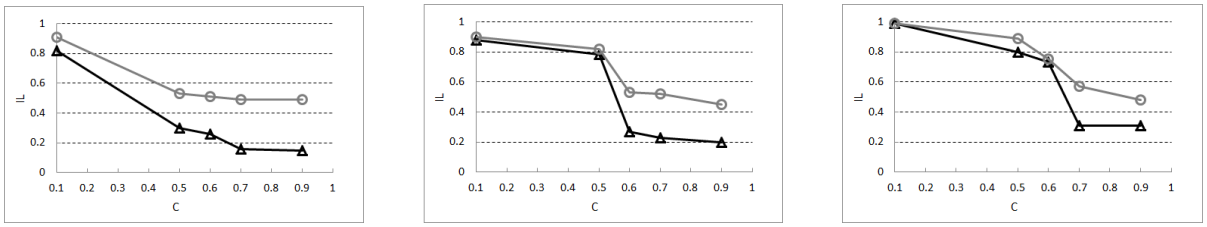
In this section, we examine the impact of anonymity threshold K, confidence threshold C, and adversary power P on the effectiveness of the $HALT$ method. We compare our method with a modified version of $HALT$ which only employs global generalization to



(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

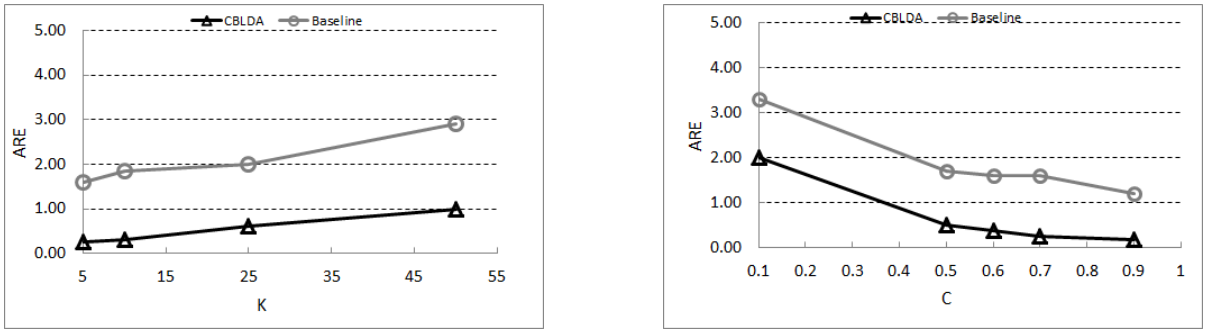


(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3



(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

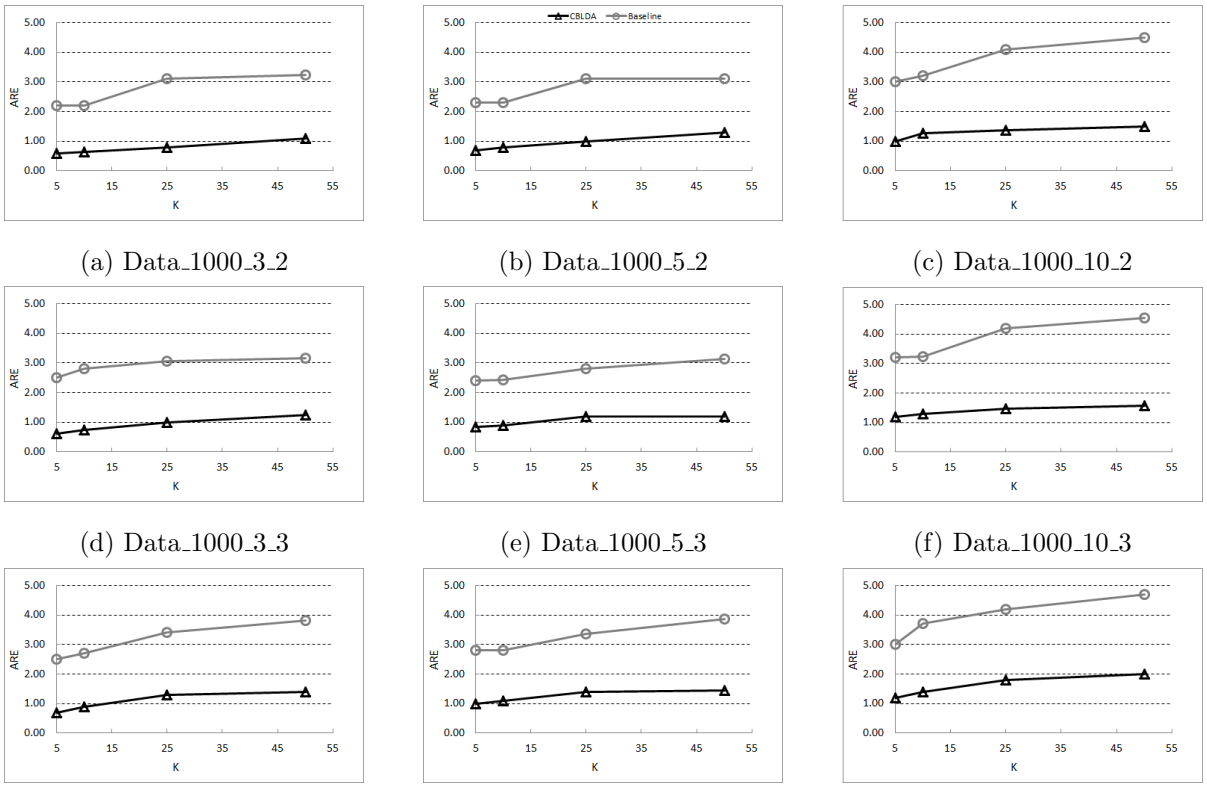
Figure 6.5: Information loss (IL) of *CBLDA* vs C on Syn-10000 (K = 5)



(a) *ARE vs K* on *CMS* data (C = 0.7) (b) *ARE vs C* on *CMS* (K = 5)

Figure 6.6: Average Relative Error (*ARE*) of *CBLDA* on *CMS* data

anonymize data, namely *GloGen*.



(a) Data_1000_3_2 (b) Data_1000_5_2 (c) Data_1000_10_2

(d) Data_1000_3_3 (e) Data_1000_5_3 (f) Data_1000_10_3

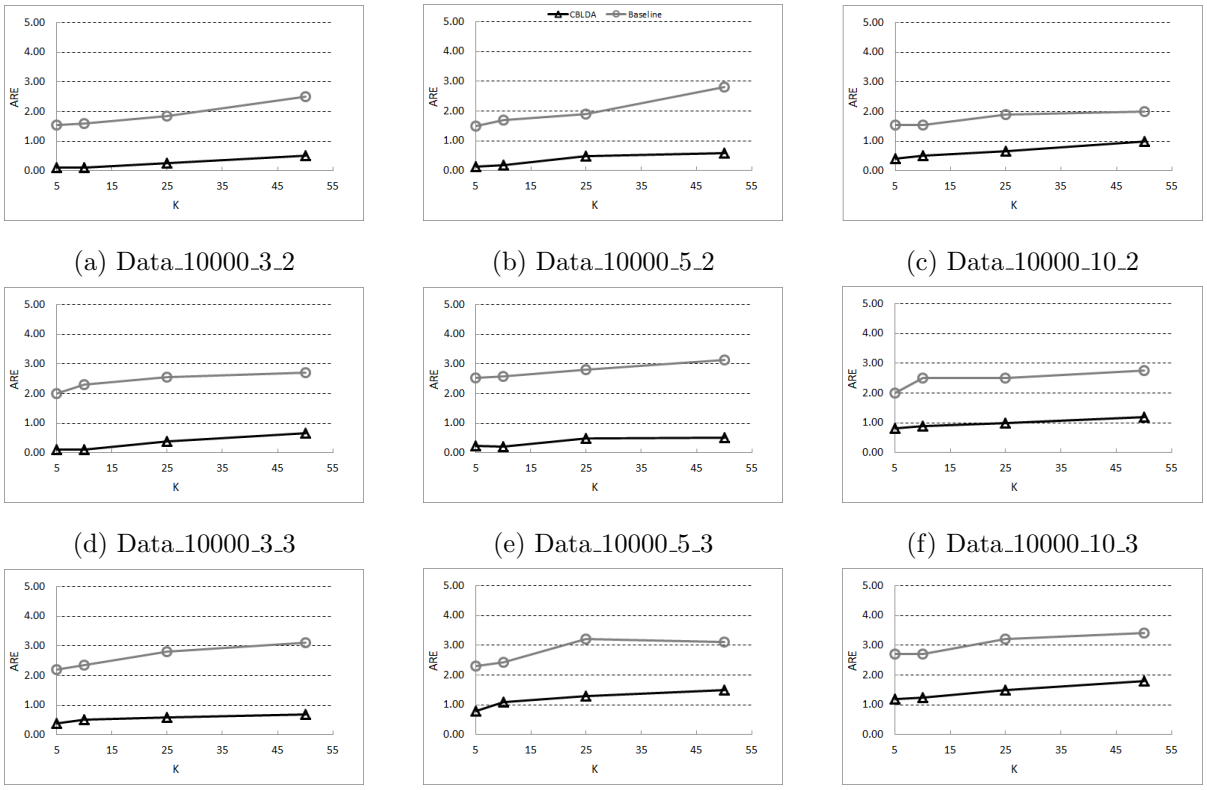
(g) Data_1000_3_4 (h) Data_1000_5_4 (i) Data_1000_10_4

Figure 6.7: Average Relative Error (ARE) of *CBLDA* vs K on Syn-1000 (C = 0.7)

6.4.1 Information Loss

Impact of K. The information loss of $HALT$ and *GloGen* on CMS data and synthetic data for $C = 0.7$, $P = 3$, and K from 5 to 50 is shown in Figures 6.11a, 6.12, and 6.13. Overall, as K increases, information loss of $HALT$ and *GloGen* algorithms increases. However, it can be seen that in some cases, information loss of $HALT$ decreases for a higher value of K, for example, on Data_1000_10_3 in Figure 6.12f. This is due to the fact that $HALT$ is a greedy algorithm and typically produces sub-optimal anonymizations.

Interestingly, the information loss incurred by $HALT$ on all datasets is comparable with the results of $CBLDA$ on the same datasets. Considering the fact that $HALT$ employs global generalization and suppression, these results imply the effectiveness of our assumption about bounding an adversary's background knowledge by P.



(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3

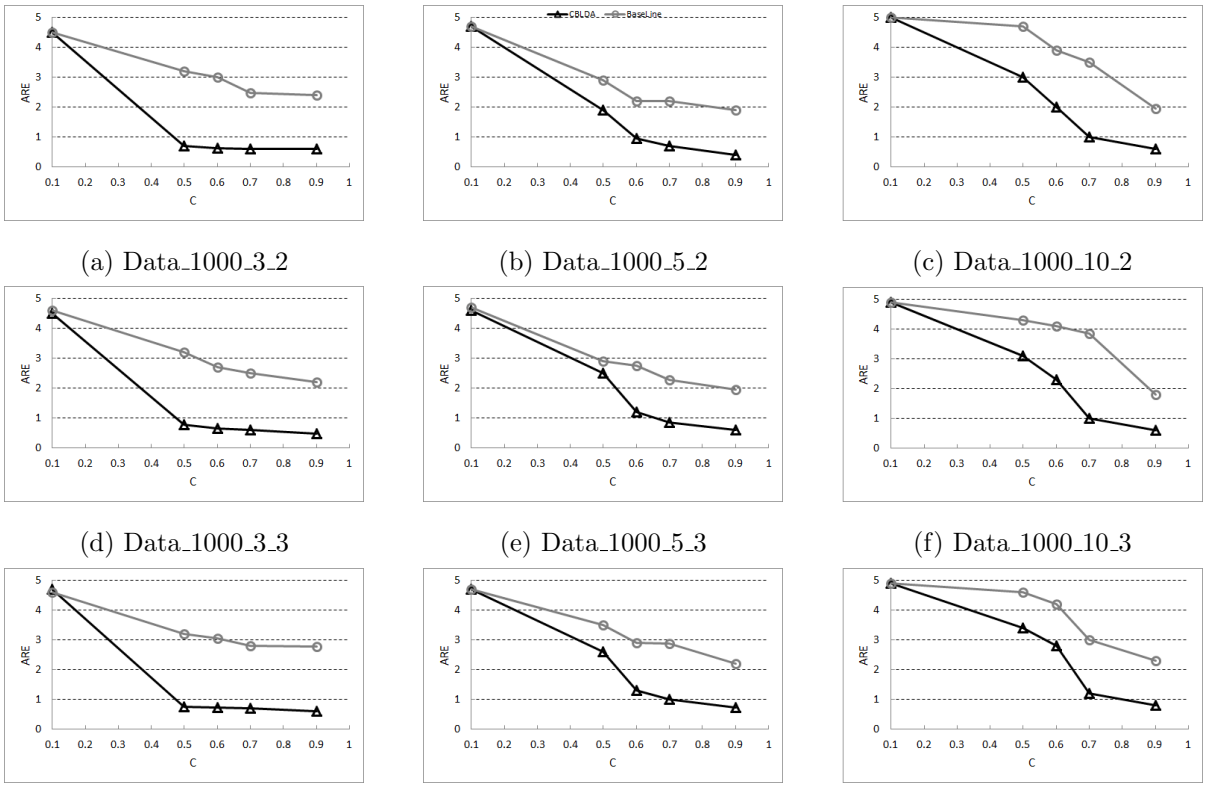
(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

Figure 6.8: Average Relative Error of *CBLDA* vs K on Syn-10000 (C = 0.7)

$HALT$, on average, incurs 54% less information loss than *GloGen* on CMS data. On Syn-1000 datasets and Syn-10000 datasets the information loss of $HALT$ is, on average, at least 34% and 45% less than *GloGen*, respectively. These results indicate the benefits of combining generalization and suppression in an anonymization process. In fact, suppression removes outliers from data and therefore a lower level of generalization would be required to satisfy privacy constraints.

Impact of C. The results for varying C, $K = 5$, and $P = 3$ are reported in Figures 6.11b, 6.14, and 6.15. As can be seen, when C increases, information loss decreases on all datasets. This is due to the fact that larger values of C lead to a smaller number of minimal inference channels in a dataset and as a result less data distortion is required to eliminate extracted inference channels.

Impact of P. To study the impact of the adversary's background knowledge on data



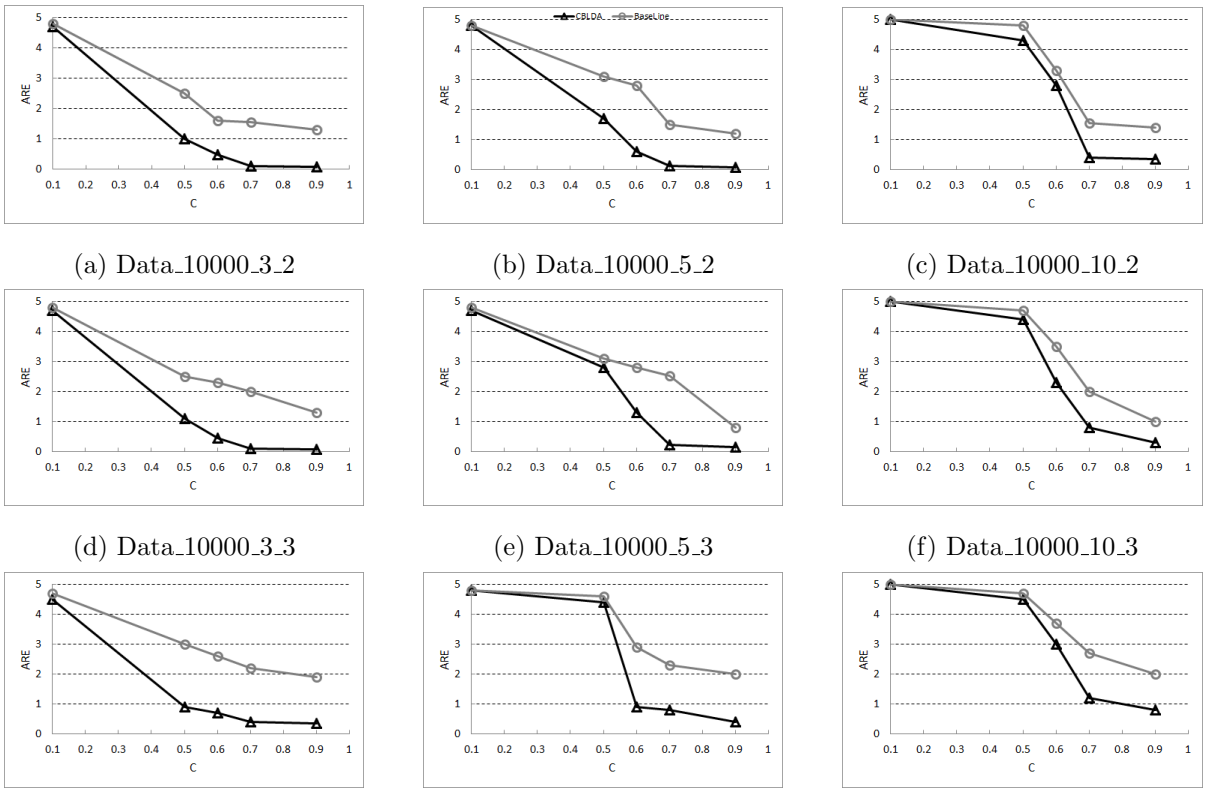
(a) Data_1000_3_2 (b) Data_1000_5_2 (c) Data_1000_10_2

(d) Data_1000_3_3 (e) Data_1000_5_3 (f) Data_1000_10_3

(g) Data_1000_3_4 (h) Data_1000_5_4 (i) Data_1000_10_4

Figure 6.9: Average Relative Error (*ARE*) of *CBLDA* vs C on Syn-1000 (K = 5)

utility, we varied the value of adversary's power P from 2 to 5 while fixing $K = 5$ and $C = 0.7$. Figures 6.16, 6.17, and 6.18 report the information loss incurred by $HALT$ and *GloGen* on CMS data, Syn-1000 datasets, and Syn-10000 datasets. As can be seen, the information loss at $P = 2$ and $P = 3$ is not high. This implies that the number of minimal inference channels of size 2 and 3 in the data is not high and this has led to less data distortion. As P increases, the number of extracted inference channels increases and, thus, more generalization and suppression are required to eliminate inference channels. In fact, a higher value of P means an adversary with more background knowledge. Therefore, more data distortion is required to protect privacy of individuals in data.

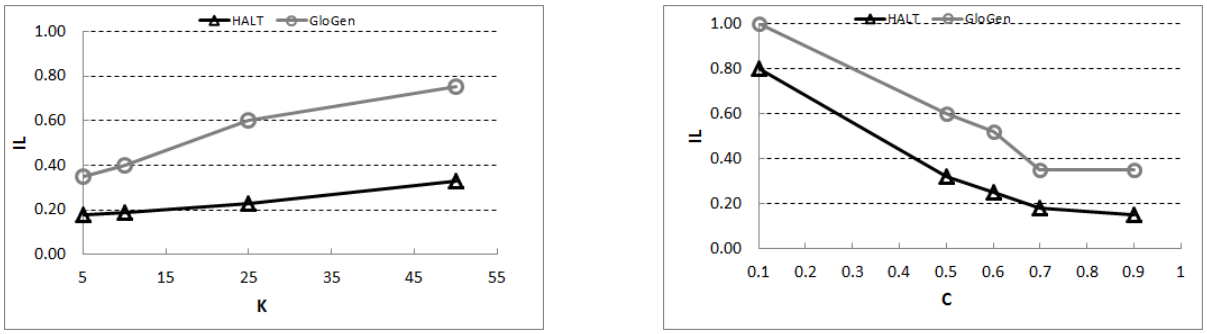


(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3

(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

Figure 6.10: Average Relative Error (*ARE*) of *CBLDA* vs C on Syn-10000 (K = 5)

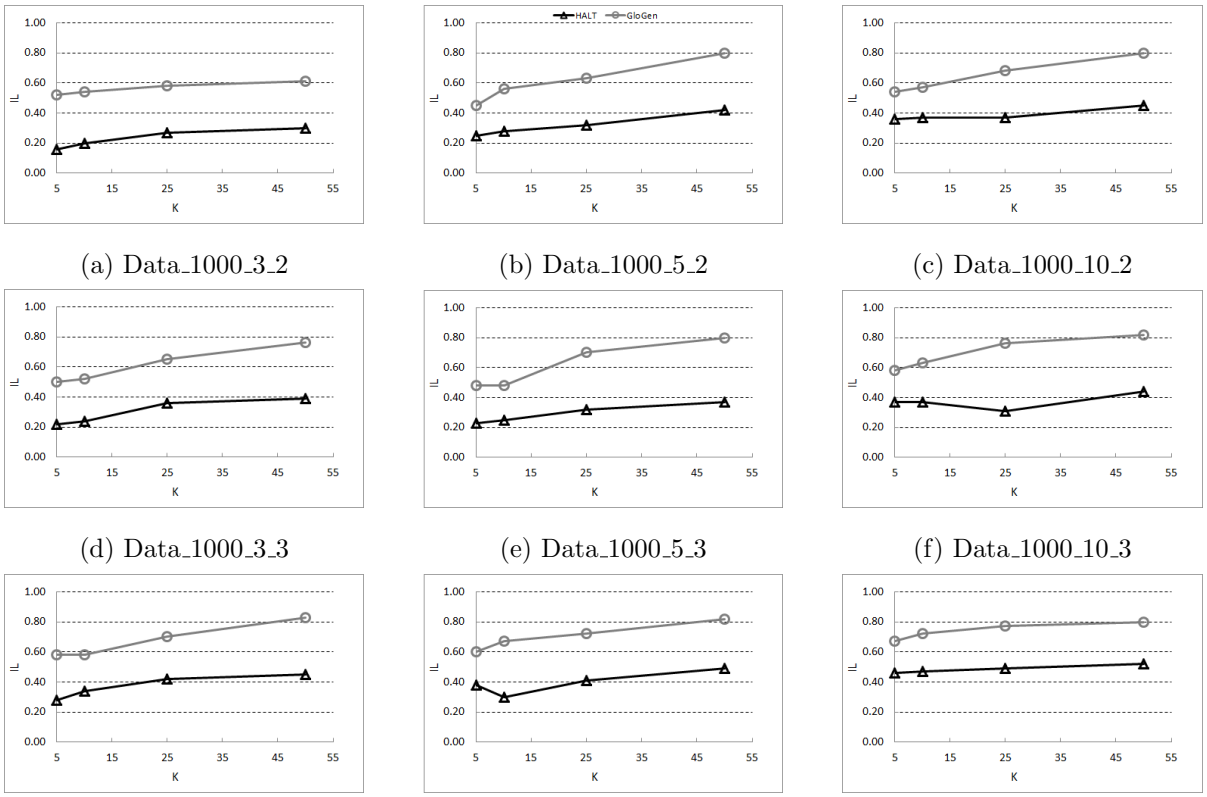


(a) IL *vs* K on CMS ($C = 0.7$, $P = 3$) (b) IL *vs* C on CMS ($K = 5$, $P = 3$)

Figure 6.11: Information loss (IL) of *HALT* on *CMS* data

6.4.2 Query Answering Accuracy

Impact of K. Figures 6.19a, 6.20, and 6.21 show *ARE* values of *HALT* and *GloGen* for *CMS* data and synthetic datasets, where K varies from 5 to 50, $C = 0.7$ and $P = 3$.



(a) Data_1000_3_2 (b) Data_1000_5_2 (c) Data_1000_10_2

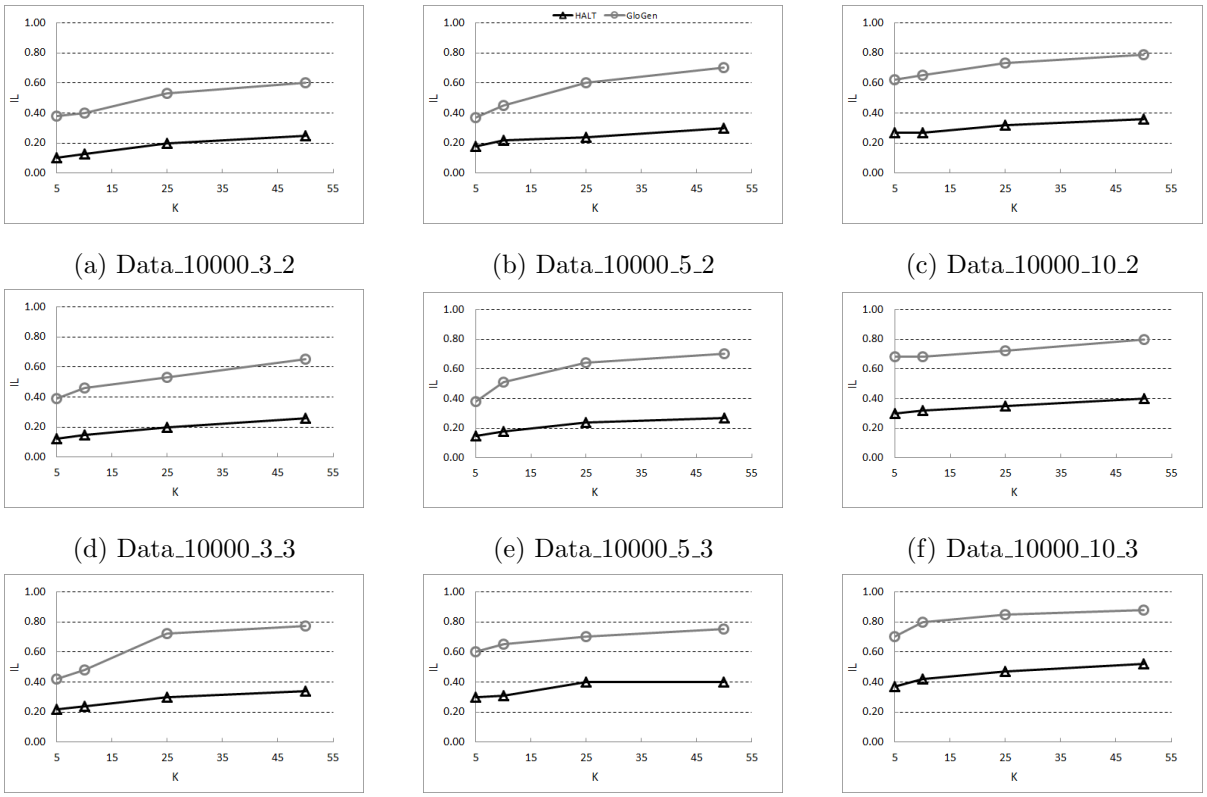
(d) Data_1000_3_3 (e) Data_1000_5_3 (f) Data_1000_10_3

(g) Data_1000_3_4 (h) Data_1000_5_4 (i) Data_1000_10_4

Figure 6.12: Information loss (IL) of $HALT$ vs K on Syn-1000 ($C = 0.7$, $P = 3$)

As is expected, increasing K leads to less accurate query answering as more information loss incurs on data. However, even for a large value of K, $HALT$ incurs a fairly low ARE on CMS data and all synthetic datasets. For instance, for $K = 50$, ARE for CMS data is 1.6, and for Syn-1000 datasets and Syn-10000 datasets is, on average, 1.96 and 1.58, respectively. Moreover, $HALT$ outperforms *GloGen* in accurately answering queries. More precisely, $HALT$, is 68%, on average, more accurate than *GloGen* in answering queries on CMS data. On Syn-1000 datasets and Syn-10000 datasets, $HALT$ is at least 47% and 45% more accurate than *GloGen* in query answering, on average. These results show the effectiveness of our proposed method.

Impact of C. In figures 6.19b, 6.22, and 6.23 the ARE values of $HALT$ and *GloGen* for CMS and synthetic datasets under varying C from 0.1 to 0.9 with $K = 5$



(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

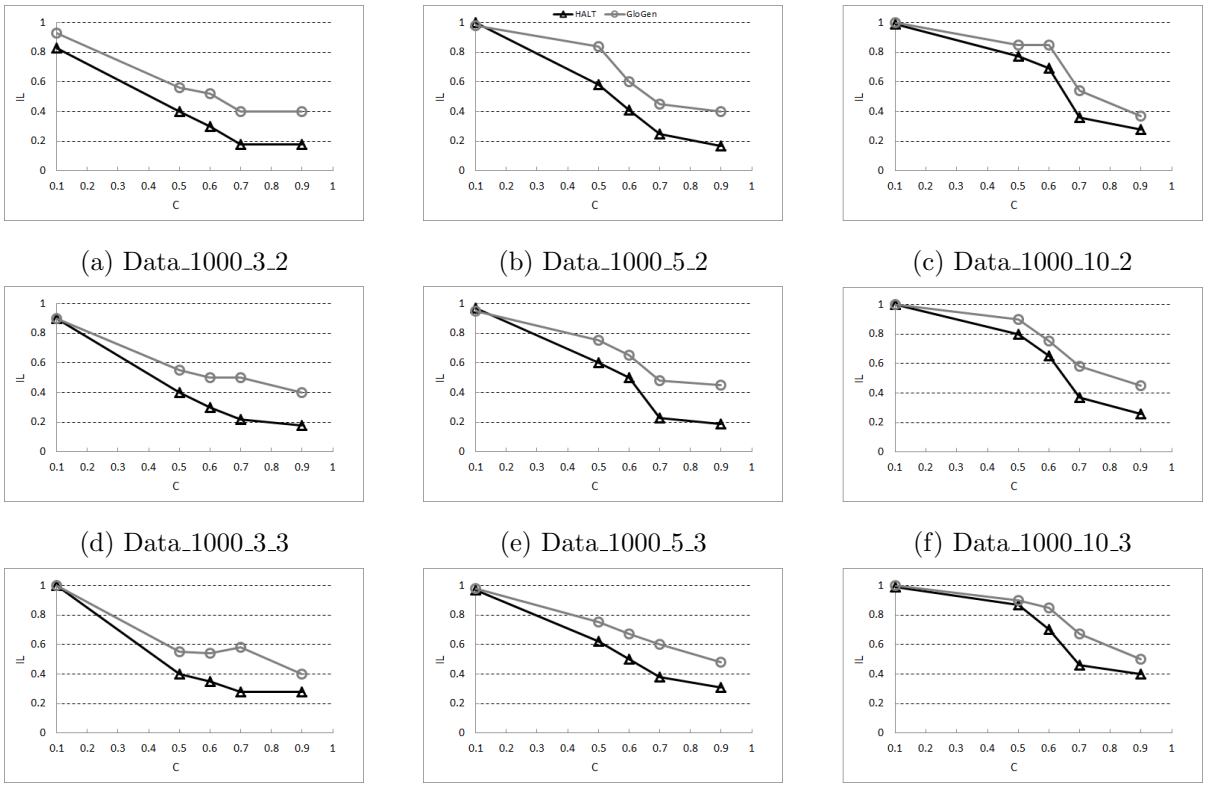
(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3

(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

Figure 6.13: Information loss (IL) of $HALT$ vs K on Syn-10000 ($C = 0.7$, $P = 3$)

and $P = 3$ is shown. As can be seen when C increases, ARE decreases. As we discussed before, if C is greater, there is a smaller requirement for privacy protection for attribute disclosure. Thus, fewer generalization and suppression operations are applied on data. The lower the number of generalization and suppression operations applied, the higher the data utility is preserved. As a result, queries which are issued on anonymized data can be answered more accurately.

Impact of P. The ARE values for CMS data and synthetic datasets are shown in figures 6.24, 6.25, and 6.26 where $K = 5$, $C = 0.7$, and P is changed from 2 to 5. As P increases, data distortion increases due to trade-off between data utility and stronger privacy protection which is enforced by P. As a result, we observe an increase in ARE values as P increases. ARE values for smaller values of P are fairly low. However,



(a) Data_1000_3_2 (b) Data_1000_5_2 (c) Data_1000_10_2

(d) Data_1000_3_3 (e) Data_1000_5_3 (f) Data_1000_10_3

(g) Data_1000_3_4 (h) Data_1000_5_4 (i) Data_1000_10_4

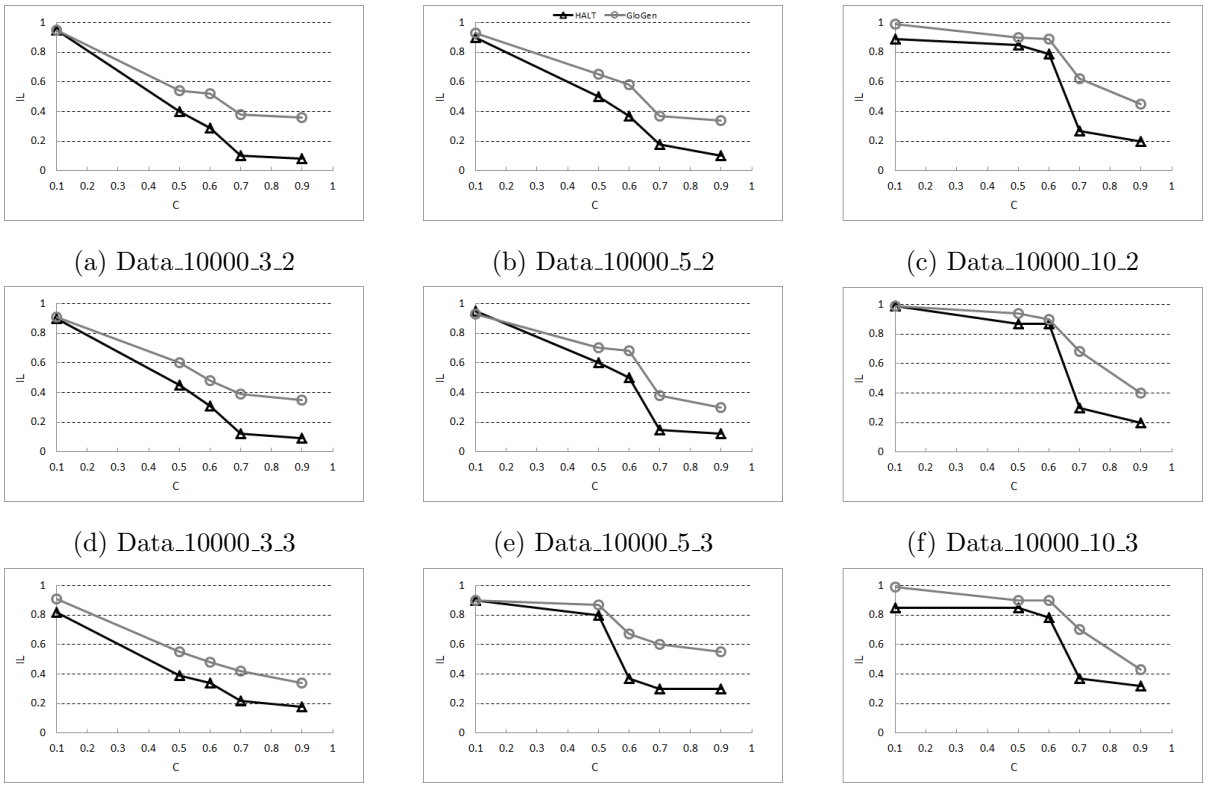
Figure 6.14: Information loss (IL) of $HALT$ vs C on Syn-1000 ($K = 5$, $P = 3$)

even for larger values of P, ARE values do not exceed 3. These results suggest the effectiveness of $HALT$ in accurate query answering.

6.5 Summary

In this chapter, we presented extensive experiments to evaluate the ability of our proposed algorithms to anonymize data while preserving data utility. We evaluated our algorithms in terms of general information loss and Average Relative Error on CMS data as well as a set of synthetic datasets. Our experimental results can be summarized as follows:

- We evaluated the effectiveness of $CBLDA$ increased by varying the value of anonymity threshold K and confidence threshold C. We observed that the information loss of $CBLDA$ by increasing K for a fixed value of C. This is due to the fact that,



(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3

(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

Figure 6.15: Information loss (IL) of $HALT$ vs C on Syn-10000 ($K = 5$, $P = 3$)

more generalization and suppression are required to ensure that each cluster has at least K indistinguishable records. We also observed that, when the number of quasi-identifiers increases, the information loss increases for any value of K. This can be explained by the fact that more attributes should be considered during dynamic programming to align two sequences. We noticed the same trend when the average number of events per sequence is increased.

We also evaluated the performance of our algorithm against various values of confidence threshold C. We observed that by increasing C, the information loss drops significantly. This is due to the fact that as C increases, privacy restrictions decrease and a smaller amount of generalization and suppression will be needed to anonymize data. Interestingly, information loss using $CBLDA$ on all datasets were fairly low.

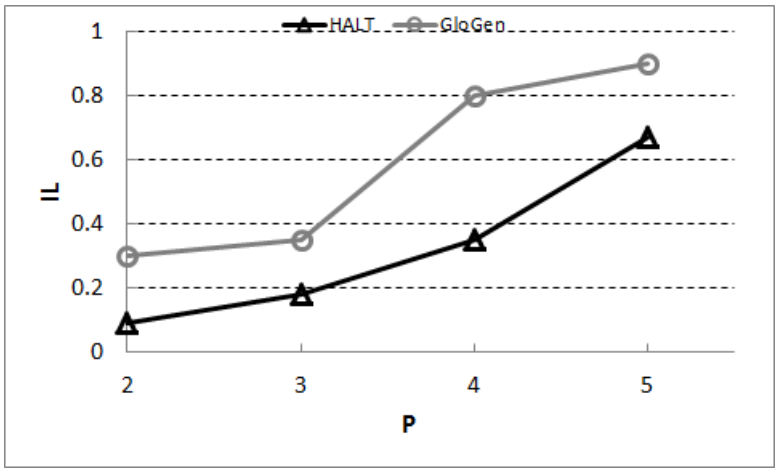


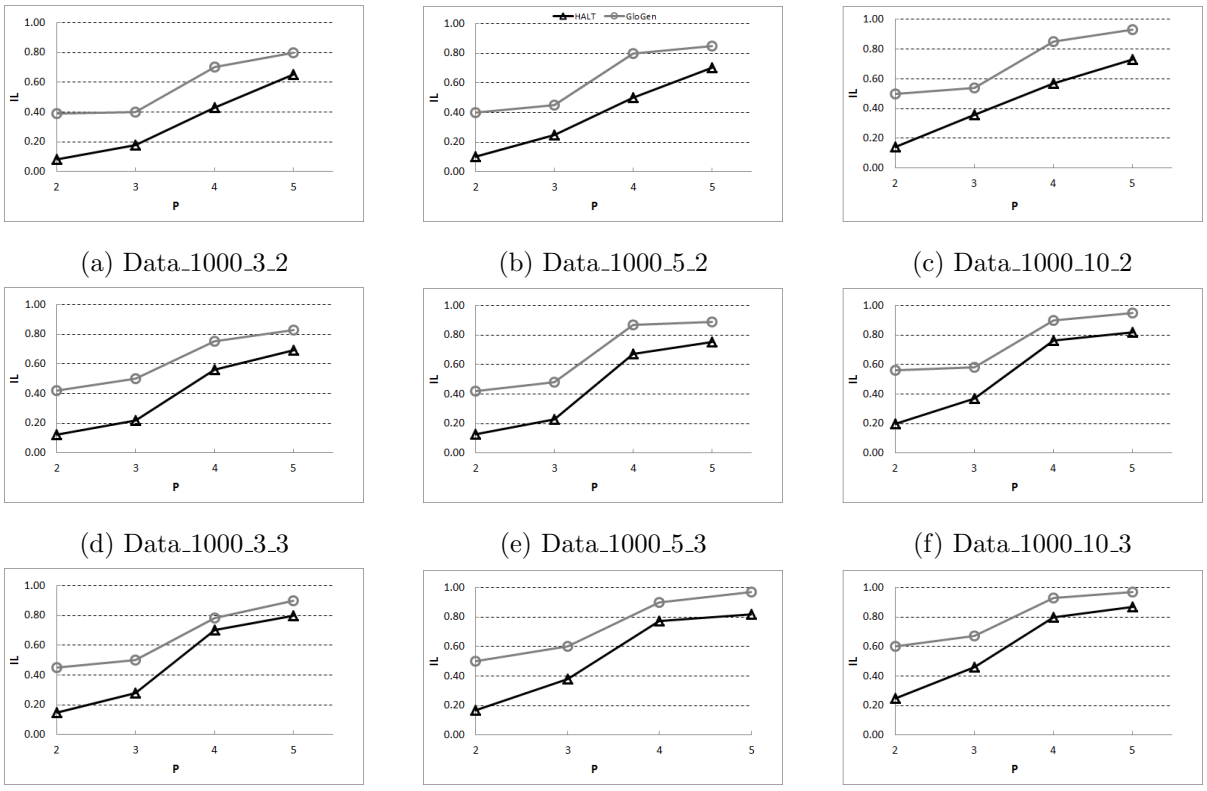
Figure 6.16: Information loss (IL) of $HALT$ vs P on CMS data ($K = 5$, $C = 0.7$)

In addition to information loss, we also evaluated our algorithm in terms of average relative error by varying K and C. We observed that the value of ARE increases when K increases. This is because for higher values of K, more generalization and suppression is applied to data and, as a result, the uncertainty in estimating the answer of an aggregate query which is issued on anonymized data increases. We noticed that ARE is fairly insensitive to changes in C for $C \geq 0.5$.

In general, information loss and ARE of $CBLDA$ for all datasets are fairly low. This is mostly due to the fact that $CBLDA$ groups similar sequences together and, therefore, less generalization and suppression will be required in each cluster to make sequences identical.

- We also evaluated the $HALT$ algorithm by varying the anonymity threshold K, confidence threshold C, and adversary's power P. Interestingly the information loss incurred by $HALT$ on all datasets is comparable with the results of $CBLDA$ on the same dataset. Considering the fact that we employ global generalization and global suppression in $HALT$, the achieved results demonstrate the effectiveness of bounding an adversary's background knowledge by P.

  We also evaluated the average relative error which incurred by $HALT$ for various values of K, C, and P. In general, increasing K and P, and decreasing C leads to



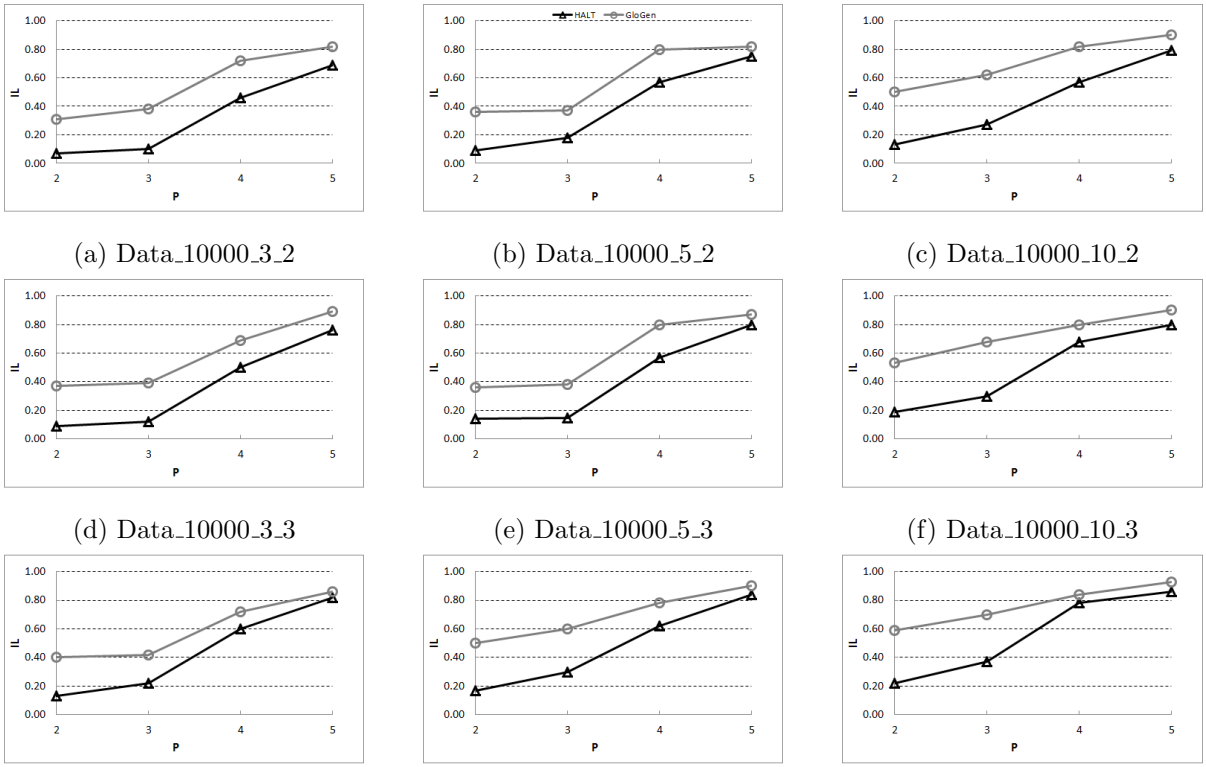
(a) Data_1000_3_2 (b) Data_1000_5_2 (c) Data_1000_10_2

(d) Data_1000_3_3 (e) Data_1000_5_3 (f) Data_1000_10_3

(g) Data_1000_3_4 (h) Data_1000_5_4 (i) Data_1000_10_4

Figure 6.17: Information loss (IL) of $HALT$ vs P on Syn-1000 (K =5, C = 0.7)

less accurate query answering as more information loss incurs on data. However, all results are fairly low which demonstrates the effectiveness of $HALT$ in accurate query answering.

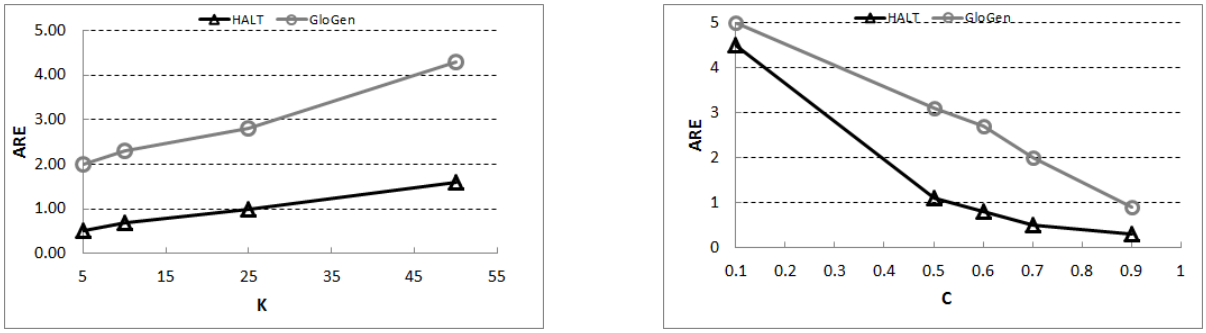


(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3

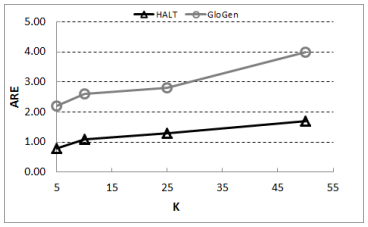
(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

Figure 6.18: Information loss (IL) of $HALT$ vs P on Syn-10000 ($K = 5$, $C = 0.7$)

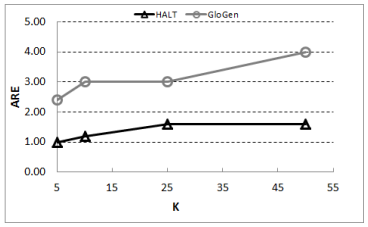


(a) ARE *vs* K on CMS ($C = 0.7$, $P = 3$) (b) ARE *vs* C on CMS ($K = 5$)

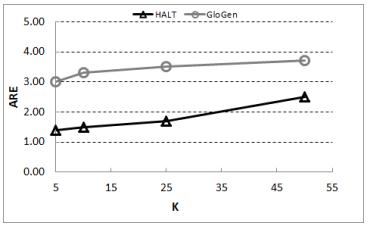
Figure 6.19: Average Relative Error (ARE) of $HALT$ on CMS data



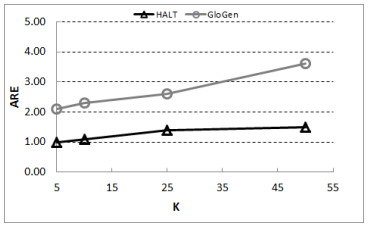
(a) Data_1000_3_2



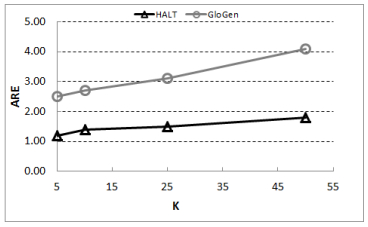
(b) Data_1000_5_2



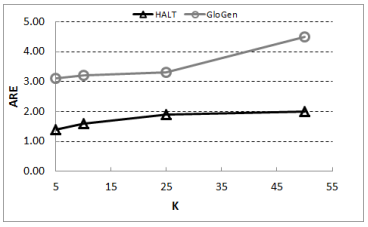
(c) Data_1000_10_2



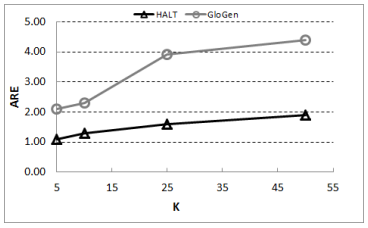
(d) Data_1000_3_3



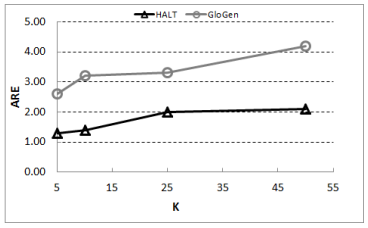
(e) Data_1000_5_3



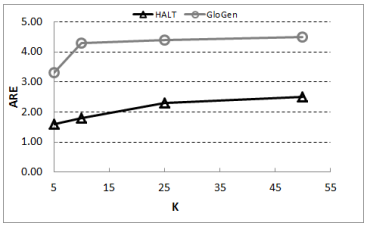
(f) Data_1000_10_3



(g) Data_1000_3_4

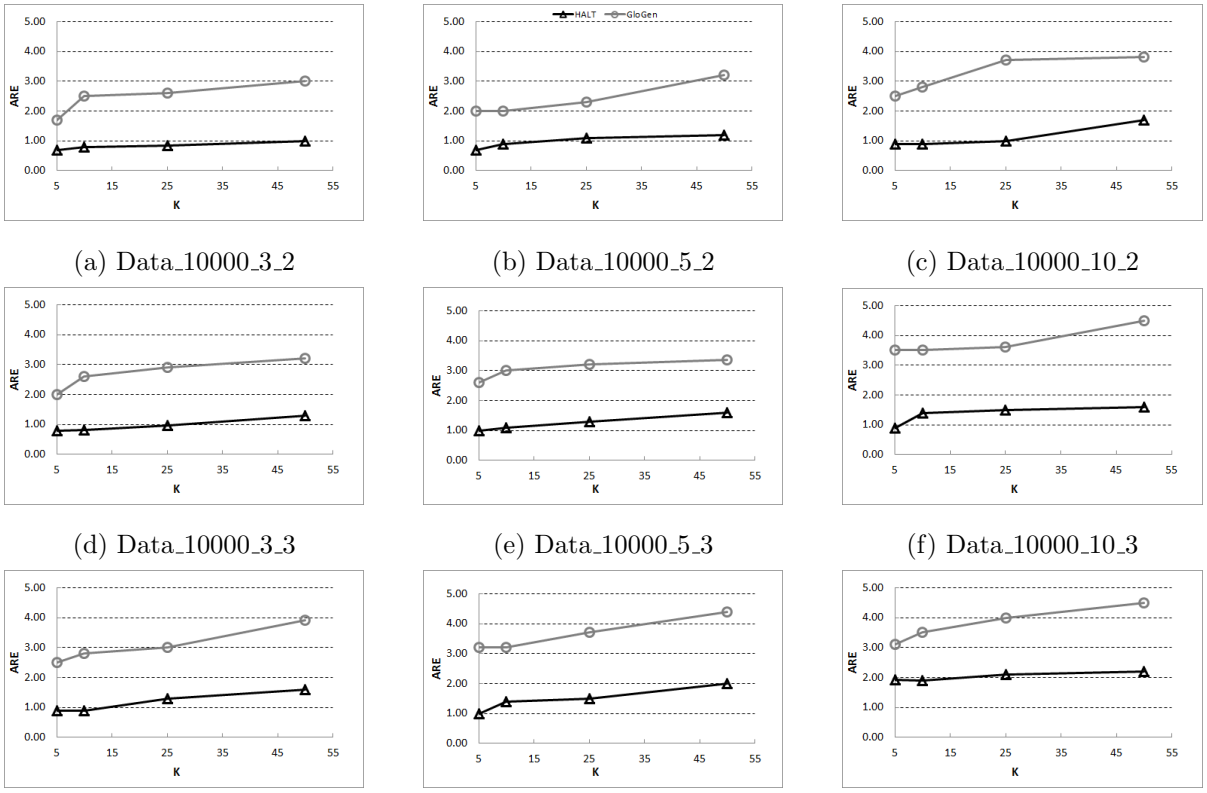


(h) Data_1000_5_4



(i) Data_1000_10_4

Figure 6.20: Average Relative Error (ARE) of $HALT$ vs K on Syn-1000 ($C = 0.7$, $P = 3$)

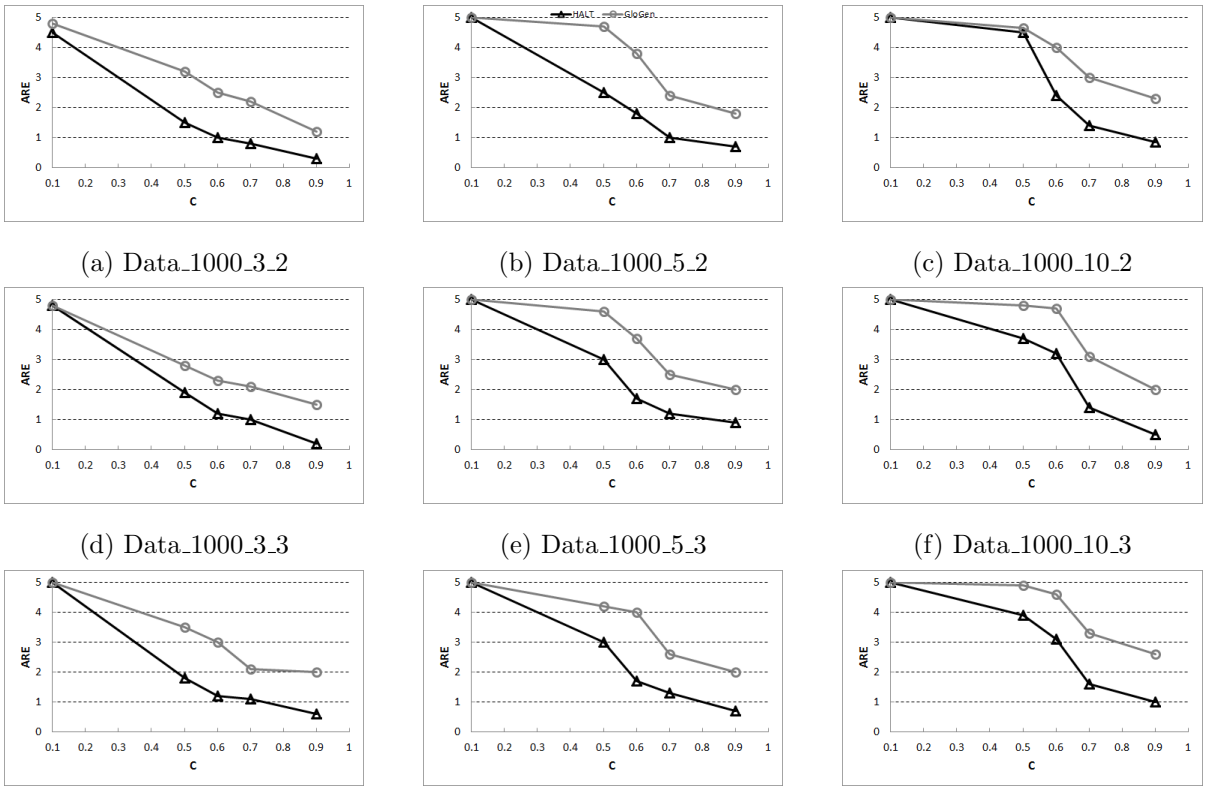


(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3

(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

Figure 6.21: Average Relative Error (ARE) of $HALT$ vs K on Syn-10000 ($C = 0.7$, $P = 3$)

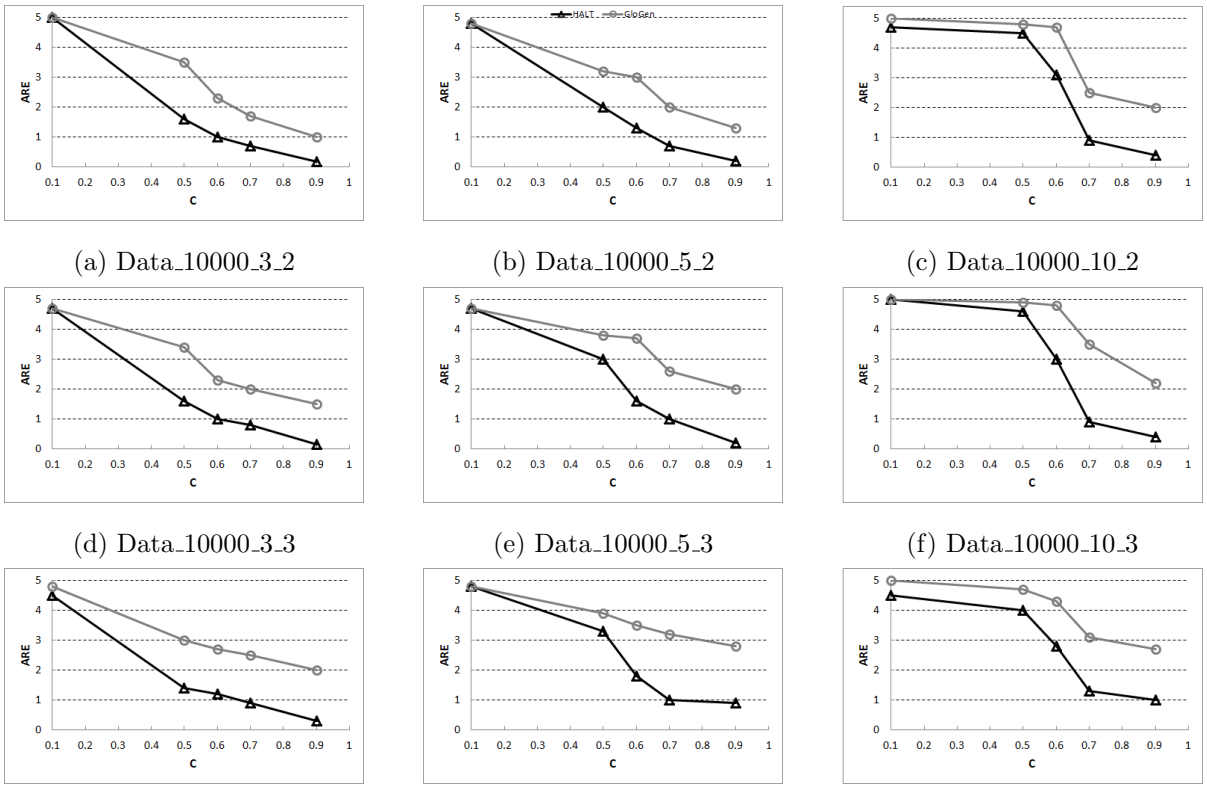


(a) Data_1000_3_2 (b) Data_1000_5_2 (c) Data_1000_10_2

(d) Data_1000_3_3 (e) Data_1000_5_3 (f) Data_1000_10_3

(g) Data_1000_3_4 (h) Data_1000_5_4 (i) Data_1000_10_4

Figure 6.22: Average Relative Error (ARE) of $HALT$ vs C on Syn-1000 ($K = 5$, $P = 3$)



(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3

(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

Figure 6.23: Average Relative Error (ARE) of $HALT$ vs C on Syn-10000 ($K = 5$, $P = 3$)

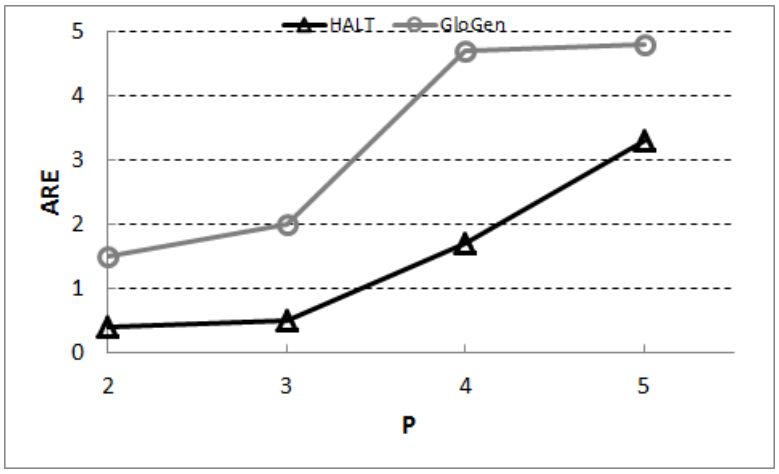
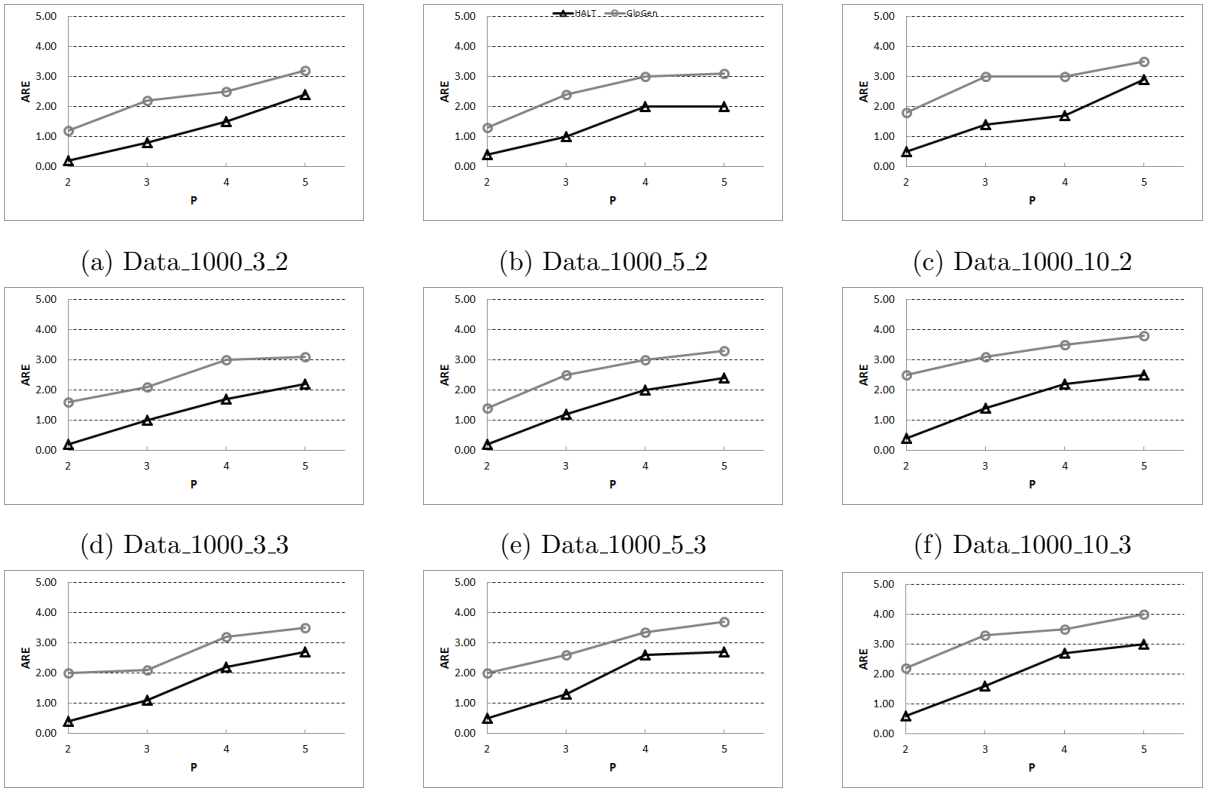


Figure 6.24: AVerage Raletive Error (ARE) of $HALT$ vs P on CMS data ($K = 5$, $C = 0.7$)

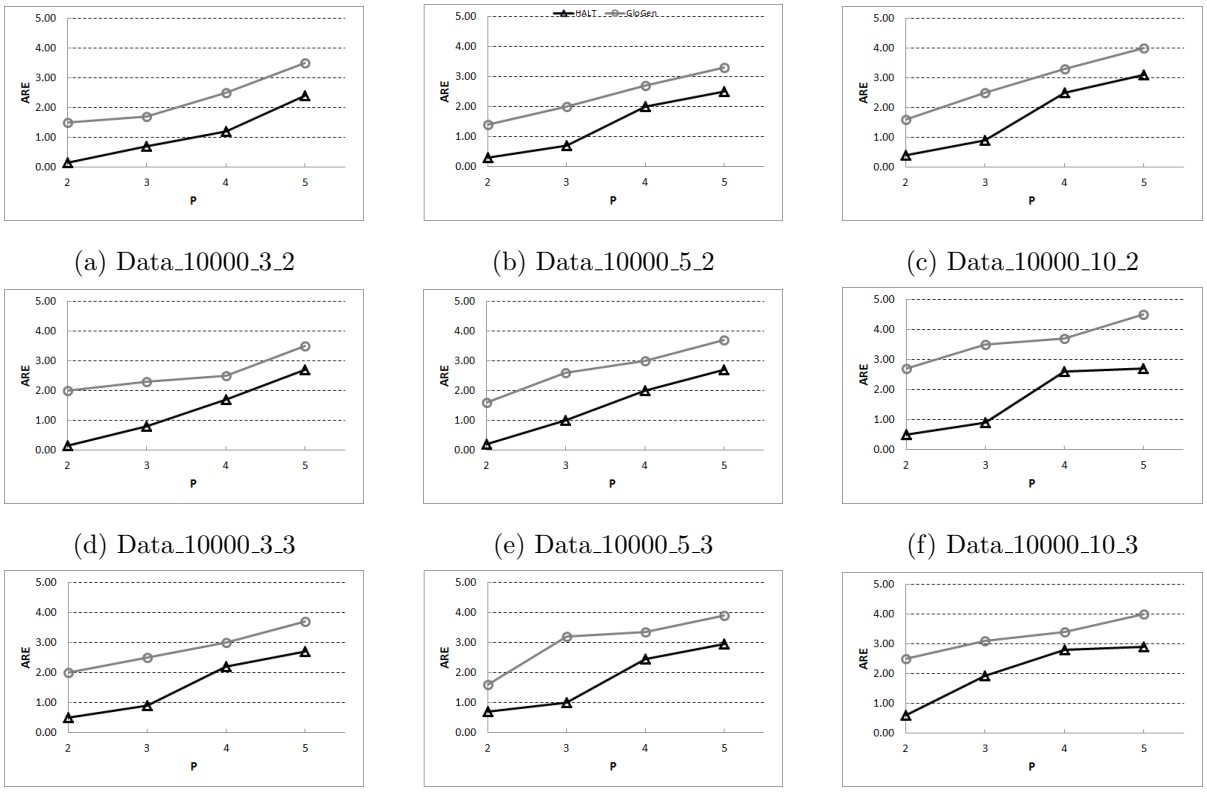


(a) Data_1000_3_2 (b) Data_1000_5_2 (c) Data_1000_10_2

(d) Data_1000_3_3 (e) Data_1000_5_3 (f) Data_1000_10_3

(g) Data_1000_3_4 (h) Data_1000_5_4 (i) Data_1000_10_4

Figure 6.25: Average Relative Error (ARE) of $HALT$ vs P on Syn-1000 ($K = 5$, $C = 0.7$)



(a) Data_10000_3_2 (b) Data_10000_5_2 (c) Data_10000_10_2

(d) Data_10000_3_3 (e) Data_10000_5_3 (f) Data_10000_10_3

(g) Data_10000_3_4 (h) Data_10000_5_4 (i) Data_10000_10_4

Figure 6.26: Average Relative Error (ARE) of $HALT$ vs P on Syn-10000 ($K = 5$, $C = 0.7$)

Chapter 7

Conclusion and Future Work

Privacy preserving data publishing is an important research area which studies the techniques to eliminate privacy threats from data while preserving useful information for data mining and knowledge discovery approaches. It provides methods and tools for transforming data to satisfy both privacy and utility constraints. In this thesis, we studied the problem of privacy-preserving longitudinal data publishing. Longitudinal data contain information of multiple observations or events related to an individual collected over time. A good example of longitudinal data is successive hospitalizations of patients in a hospital. In fact, all clinical data found in EMRs are often longitudinal by nature and despite the high sensitivity of information which exist in longitudinal health data, no effective method has been proposed so far to anonymize longitudinal data to prevent both identity disclosure and attribute disclosure. To respond to privacy concerns which have been raised in sharing longitudinal data, in this thesis, we proposed an efficient and effective novel privacy-preserving framework for longitudinal data publishing which prevents both identity disclosure and attribute disclosure. We summarize the major contributions of this thesis as follows:

- In Chapter 3, we presented a new taxonomy of privacy preserving data publishing approaches in the literature. We studied privacy preserving approaches for publishing data with respect to two types of disclosure: identity disclosure and attribute

disclosure. In each of these categories, we discussed different types of potential background knowledge of adversaries as well as the potential privacy attacks based on such knowledge. Then, we reviewed the proposed solutions in the literature to prevent the aforementioned privacy attacks. We classified these solutions into three categories based on the structure of the data being de-identified, including relational data, transaction data, and trajectory and sequence data.

- In Chapter 4, we proposed the $(K, C)^P$-privacy model to prevent identity disclosure and attribute disclosure in publishing longitudinal data in the presence of an adversary with power P. We developed an efficient hybrid anonymization algorithm, called $HALT$, which employs global generalization and global suppression to anonymize longitudinal data to satisfy $(K, C)^P$-privacy while preserving data utility as much as possible. Through extensive experiments we showed that our proposed method incurs low information loss while preserving $(K, C)^P$-privacy.

- In Chapter 5, we proposed a clustering-based anonymization algorithm using sequence alignment and hierarchical agglomerative clustering techniques to anonymize longitudinal data. We formulated a new privacy notion for longitudinal data, called (K, C)-privacy, to prevent both identity disclosure and attribute disclosure. This model does not make any assumption about the background knowledge of adversaries and preserves privacy with respect to an adversary who may know any number of quasi-identifiers. Experimental results suggest that the proposed algorithm can efficiently and effectively anonymize longitudinal data with low information loss.

We believe that this thesis opens up several important research directions which can be investigated in the future. One possible extension of our proposed framework which can be improved in future works is our assumption about the goal of data publication. We assumed that the data publisher does not know the purpose of data usage and anonymizes data by minimizing data distortion for general data analysis purposes. In future research

we can consider the case of publishing data for a specific data mining task such as classification. This requires employing an appropriate anonymization cost measure to capture the utility of our algorithms for a specific data mining task.

One limitation of this thesis is that we studied the case of a single sensitive attribute in every event of a sequence. As we mentioned in Chapter 4, extending our work to prevent identity disclosure in the case of multiple sensitive attributes is trivial. However, this extension will not be so trivial for the case of attribute disclosure due to the fact that we cannot handle each sensitive attribute separately and we should consider combinations of multiple sensitive attributes' values. Addressing challenges of anonymizing longitudinal data with multiple sensitive attributes is a possible extension of this thesis.

Another future direction of this thesis would be to deal with additional individuals' information, such as another table containing demographics. In that case, we should anonymize data with respect to combinations of information in both tables which may be exploited for privacy attacks. Developing more scalable algorithms, specifically using Hadoop[1] and cloud-computing [2] for anonymizing large longitudinal data would be another interesting direction to explore.

Privacy deals with individual, cultural, and social norms. Moreover, effectiveness and efficiency of privacy regulations and tools in protecting individual privacy is very important. From these perspectives, there are some similarities between privacy and health economics [107]. Health economics deals with issues related to effectiveness, efficiency, behaviors of individuals and cultural matters in healthcare. For instance, in health economics the smoking behavior of individuals and its impact on the health of community is studied. Equivalently, behavior of individuals who interact with a system can be studied and based on that privacy policies and tools can be designed. Cultural aspects and its impact on health of a population are also studied in health economics. For example, if people in one area eat, culturally, more sea food, then its impact on population health

[1] http://hadoop.apache.org/

[2] http://en.wikipedia.org/wiki/Cloud˙computing

can be studied in the health economics. The results of such studies can be employed, for instance, to set regulations as well as health insurance policies. Similarly, a specific group of people may be culturally sensitive to privacy violations and studying such cases is necessary to develop effective privacy tools. Considering these similarities between privacy and health economics, one possible direction for privacy research can be to model privacy in terms of models in health economics.

Bibliography

[1] Abul, O., Bonchi, F., and Nanni, M. Never walk alone: Uncertainty for anonymity in moving objects databases. In *IEEE International Conference on Data Engineering (ICDE)* (2008), G. Alonso, J. A. Blakeley, and A. L. P. Chen, Eds., IEEE, pp. 376–385.

[2] Aggarwal, C. On k-anonymity and the curse of dimensionality. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases* (2005), pp. 901–909.

[3] Aggarwal, C. C., and Yu, P. S. A framework for condensation-based anonymization of string data. *Data Mining and Knowledge Discovery 16* (June 2008), 251–275.

[4] Aggarwal, C. C., and Yu, P. S. On static and dynamic methods for condensation-based privacy-preserving data mining. *ACM Transactions on Database Systems 33* (March 2008), 1–39.

[5] Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., and Zhu, A. Anonymizing tables. In *Database Theory - ICDT 2005*, vol. 3363 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005, pp. 246–258.

[6] Agrawal, R., and Srikant, R. Privacy-preserving data mining. *ACM SIGMOD Record 29* (May 2000), 439–450.

[7] ALBERT, P. S. Longitudinal data analysis (repeated measures) in clinical trials. *Statistics in Medicine 18*, 13 (1999), 1707–1732.

[8] Associated Press. Amazon knows who you are. `http://www.wired.com/techbiz/media/news/2005/03/67034`, May 2005.

[9] Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. Sequential pattern mining using a bitmap representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2002), ACM, pp. 429–435.

[10] Backstrom, L., Dwork, C., and Kleinberg, J. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW '07: Proceedings of the 16th international conference on World Wide Web* (2007), pp. 181–190.

[11] Barbaro, M., and Zeller, T. A face is exposed for AOL searcher no. 4417749. *New York Times*, August 2006.

[12] Bayardo, R. J., and Agrawal, R. Data privacy through optimal k-anonymization. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering* (2005), pp. 217–228.

[13] Brand, R. Microdata protection through noise addition. In *Inference Control in Statistical Databases, From Theory to Practice* (London, UK, UK, 2002), Springer-Verlag, pp. 97–116.

[14] Brankovic, L., and Estivill-castro, V. Privacy issues in knowledge discovery and data mining. In *Proceedings of Australian Institute of Computer Ethics Conference (AICEC99* (1999), pp. 89–99.

[15] Burdick, D., Calimlim, M., and Gehrke, J. Mafia: a maximal frequent itemset algorithm for transactional databases. In *Data Engineering, 2001. Proceedings. 17th International Conference on* (2001), pp. 443–452.

[16] Byun, J.-W., Kamra, A., Bertino, E., and Li, N. Efficient k-anonymity using clustering technique. *CERIAS Tech Report, Purdue University*, 2006.

[17] Canadian Institutes for Health Research. Guidelines for protecting privacy and confidentiality in the design, conduct and evaluation of health research. Canadian Institutes for Health Research, 2004.

[18] Cao, J., Karras, P., Raïssi, C., and Tan, K.-L. ρ-uncertainty: inference-proof transaction anonymization. *VLDB '10: Proceedings of the 36th international conference on Very Large Data Bases 3* (2010), 1033–1044.

[19] Carlisle, D. M., Spingarn, R., and Schoenfelder, C. *California inpatient data reporting manual*, 7 ed. Office of Statewide Health Planning and Development, 2011.

[20] Chen, B., Kifer, D., LeFevre, K., and Machanavajjhala, A. Privacy-preserving data publishing. *Foundations and Trends in Databases 2*, 1-2 (2009), 1–167.

[21] Chen, R., Fung, B. C. M., Mohammed, N., Desai, B. C., and Wang, K. Privacy-preserving trajectory data publishing by local suppression. *Inf. Sci. 231* (2013), 83–97.

[22] Ciriani, V., di Vimercati, S. D. C., Foresti, S., and Samarati, P. k-anonymity. In *Secure Data Management in Decentralized Systems, T. Yu and s. Jajodia (Eds), Springer-verlag.* 2007, pp. 323–353.

[23] Ciriani, V., di Vimercati, S. D. C., Foresti, S., and Samarati, P. Microdata protection. In *Secure Data Management in Decentralized Systems, T. Yu and s. Jajodia (Eds), Springer-verlag*. 2007, pp. 291–321.

[24] Clifton, C., and Tassa, T. On syntactic anonymity and differential privacy. *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW) 0* (2013), 88–93.

[25] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[26] Domingo-Ferrer, J., and c Torra, V. *A quantitative comparison of disclosure control methods for microdata*. Elsevier, 2001, pp. 111–133.

[27] Domingo-Ferrer, J., and Mateo-Sanz, J. M. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Trans. on Knowl. and Data Eng. 14*, 1 (2002), 189–201.

[28] Domingo-Ferrer, J., and Torra, V. A critique of k-anonymity and some of its enhancements. In *ARES '08: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security* (2008), pp. 990–993.

[29] Domingo-Ferrer, J., and Trujillo-Rasua, R. Microaggregation- and permutation-based anonymization of movement data. *Information Sciences 208* (2012), 55–80.

[30] Dong, G., and Pei, J. vol. 33 of *Advances in Database Systems*. Springer US, 2007.

[31] Du, W., and Zhan, Z. Using randomized response techniques for privacy-preserving data mining. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2003), KDD '03, ACM, pp. 505–510.

[32] Dwork, C. Differential privacy. In *ICALP (2)* (2006), pp. 1–12.

[33] Eddy, S. What is dynamic programming? *Nature Biotechnology 22* (2004), 909–910.

[34] El Emam, K., Arbuckle, L., Koru, G., Eze, B., Gaudette, L., Neri, E., Rose, S., Howard, J., and Gluck, J. De-identification methods for open health data: The case of the heritage health prize claims dataset. *Journal of Medical Internet Research 14*, 1 (2012), 1–16.

[35] El Emam, K., Buckeridge, D. L., Tamblyn, R., Neisa, A., Jonker, E., and Verma, A. The re-identification risk of canadians from longitudinal demographics. *BMC Medical Information and Decision Making 11* (2011), 46.

[36] El Emam, K., and Dankar, F. Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association 15*, 5 (2008), 627–637.

[37] El Emam, K., Dankar, F. K., and et. al. A globally optimal k-anonymity method for the de-identification of health data. *JAMIA 16* (2009), 670–682.

[38] El Emam, K., Jonker, E., Sehatakr, M., Wunderlich, J., and Gaudette, L. Managing the risk of re-identification for public use files. Report prepared for the office of the privacy commissioner of Canada, `http://www.priv.gc.ca/resource/cp/2010-2011/p_201011_15_e.asp`, 2011.

[39] Evfimievski, A., Gehrke, J., and Srikant, R. Limiting privacy breaches in privacy preserving data mining. In *PODS '03: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (2003), pp. 211–222.

[40] Frikken, K. B., and Zhang, Y. Yet another privacy metric for publishing micro-data. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society* (2008), WPES '08, ACM, pp. 117–122.

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

[42] Fung, B. C. M., Wang, K., and Yu, P. S. Top-down specialization for information and privacy preservation. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering* (2005), pp. 205–216.

[43] Fung, B. C. M., Wang, K., and Yu, P. S. Anonymizing classification data for privacy preservation. *IEEE Trans. on Knowl. and Data Eng. 19*, 5 (2007), 711–725.

[44] Ghinita, G., Tao, Y., and Kalnis, P. On the anonymization of sparse high-dimensional data. In *ICDE '08: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering* (2008), pp. 715–724.

[45] Gionis, A., Mazza, A., and Tassa, T. k-anonymization revisited. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering* (2008), pp. 744–753.

[46] Gkoulalas-Divanis, A., and Loukides, G. Utility-guided clustering-based transaction data anonymization. *Transactions on Data Privacy 5*, 1 (2012), 223–251.

[47] Han, J., Cheng, H., Xin, D., and Yan, X. Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery 15*, 1 (2007), 55–86.

[48] Hay, M., Miklau, G., Jensen, D., Towsley, D., and Weis, P. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow. 1*, 1 (2008), 102–114.

[49] He, Y., and Naughton, J. F. Anonymization of set-valued data via top-down, local generalization. *Proceedings of the VLDB Endowment 2*, 1 (2009), 934–945.

[50] Iyengar, V. S. Transforming data to satisfy privacy constraints. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (2002), pp. 279–288.

[51] John, G. H. Behind-the-scenes data mining: a report on the KDD-98 panel. *SIGKDD Explor. Newsl. 1*, 1 (1999), 6–8.

[52] Kargupta, H., Datta, S., Wang, Q., and Sivakumar, K. On the privacy preserving properties of random data perturbation techniques. In *Proceedings of the Third IEEE International Conference on Data Mining* (2003), ICDM '03, IEEE Computer Society, pp. 99–106.

[53] Kaufman, L., and Rousseeuw, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.

[54] Kim, J. J., Winkler, W. E., and Census, B. O. T. Masking microdata files. In *Proceedings of the Survey Research Methods Section, American Statistical Association* (1995), pp. 114–119.

[55] Kulynych, J., and Korn, D. The effect of the new federal medical privacy rule on research. *The New England Journal of Medicine 346*, 3 (2002), 201–204.

[56] Kunkle, D., Zhang, D., and Cooperman, G. Mining frequent generalized itemsets and generalized association rules without redundancy. *J. Comput. Sci. Technol. 23*, 1 (2008), 77–102.

[57] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R. Incognito: efficient full-domain k-anonymity. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (2005), pp. 49–60.

[58] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R. Mondrian multidimensional K-anonymity. In *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering* (2006).

[59] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R. Workload-aware anonymization. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), pp. 277–286.

[60] Li, J., Tao, Y., and Xiao, X. Preservation of proximity privacy in publishing numerical sensitive data. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (2008), pp. 473–486.

[61] Li, N., and Li, T. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of IEEE International Conference on Data Engineering* (2007).

[62] Li, T., and Li, N. Optimal k-anonymity with flexible generalization schemes through bottom-up searching. In *Sixth IEEE International Conference on Data Mining* (2006), pp. 518–523.

[63] Li, T., and Li, N. *Injector*: Mining background knowledge for data anonymization. In *ICDE '08: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering* (2008), pp. 446–455.

[64] Lin, J.-L., and Wei, M.-C. An efficient clustering method for k-anonymization. In *Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society* (2008), PAIS '08, ACM, pp. 46–50.

[65] Liu, J., and Wang, K. Anonymizing transaction data by integrating suppression and generalization. In *PAKDD '10: Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2010), pp. 171–180.

[66] Loukides, G., and Gkoulalas-Divanis, A. Utility-preserving transaction data anonymization with low information loss. *Expert Syst. Appl. 39*, 10 (2012), 9764–9777.

[67] Loukides, G., and Gkoulalas-Divanis, A. Utility-aware anonymization of diagnosis codes. *IEEE J. Biomedical and Health Informatics 17*, 1 (2013), 60–70.

[68] Loukides, G., Gkoulalas-Divanis, A., and Malin, B. COAT: Constraint-based anonymization of transactions. Tech. Rep. arXiv:0912.2548, Dec 2009.

[69] Loukides, G., Gkoulalas-Divanis, A., and Malin, B. COAT: Constraint-based anonymization of transactions. *Knowledge and Information Systems* (2010), 1–32.

[70] Loukides, G., Gkoulalas-Divanis, A., and Shao, J. Anonymizing transaction data to eliminate sensitive inferences. In *Proceedings of the 21st international conference on Database and expert systems applications: Part I* (2010), DEXA'10, pp. 400–415.

[71] Loukides, G., Gkoulalas-Divanis, A., and Shao, J. Efficient and flexible anonymization of transaction data. *Knowledge and Information Systems 36*, 1 (2013), 153–210.

[72] Loukides, G., Gkoulalas-Divanis, A., and Shao, J. Efficient and flexible anonymization of transaction data. *Knowl. Inf. Syst. 36*, 1 (2013), 153–210.

[73] Mabroukeh, N. R., and Ezeife, C. I. A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv. 43*, 1 (2010), 3:1–3:41.

[74] Machanavajjhala, A., Gehrke, J., Kifer, D., and Venkitasubramaniam, M. *l*-diversity: privacy beyond k-anonymity. In *ICDE '06. Proceedings of the 22nd International Conference on Data Engineering, 2006.* (April 2006), pp. 24–24.

[75] Malin, B. An evaluation of the current state of genomic data privacy protection technology and a roadmap for the future. *Journal of the American Medical Informatics Association 12* (2005), 28–34.

[76] Malin, B. Re-identification of familial database records. In *Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium* (2006), pp. 524–528.

[77] Malin, B., and Sweeney, L. How (not) to protect genomic data privacy in a distributed network: Using trail re-identification to evaluate and design anonymity protection systems. *Journal of Biomedical Informatics 37* (2004), 179–192.

[78] Manning, C., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[79] Martin, D. J., Kifer, D., Machanavajjhala, A., Gehrke, J., and Halpern, J. Y. Worst-case background knowledge for privacy-preserving data publishing. In *IEEE 23rd International Conference on Data Engineering* (2007), pp. 126–135.

[80] Meyerson, A., and Williams, R. On the complexity of optimal k-anonymity. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2004), PODS '04, ACM, pp. 223–228.

[81] Mohammed, N., Fung, B., and Debbabi, M. Walking in the crowd: anonymizing trajectory data for pattern analysis. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management* (2009), pp. 1441–1444.

[82] Mohammed, N., Fung, B. C. M., Hung, P. C. K., and Lee, C. Anonymizing healthcare data: A case study on the blood transfusion service. In *Proceedings of*

the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (2009), pp. 1285–1294.

[83] Narayanan, A., and Shmatikov, V. De-anonymizing social networks. In *SP '09: Proceedings of the 2009 30th IEEE Symposium on Security and Privacy* (2009), pp. 173–187.

[84] Needleman, S. B., and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol. 48* (1970), 443–453.

[85] Nergiz, M., Atzori, M., Saygin, Y., and Güç, B. Towards trajectory anonymization: A generalization-based approach. *Trans. Data Privacy 2*, 1 (2009), 47–75.

[86] Nergiz, M. E., Clifton, C., and Nergiz, A. E. Multirelational k-anonymity. *IEEE Trans. on Knowl. and Data Eng. 21*, 8 (2009), 1104–1117.

[87] Øhrn, A., and Ohno-Machado, L. Using boolean reasoning to anonymize databases. *Artificial Intelligence in Medicine 15*, 3 (1999), 235–254.

[88] Orwell, G. *1984*, Centennial. ed. Tandem Library, 1950.

[89] Pensa, R., Monreale, A., Pinelli, F., and Pedreschi, D. Pattern-preserving k-anonymization of sequences and its application to mobility data mining. In *PiLBA* (2008).

[90] Plantevit, M., Laurent, A., Laurent, D., Teisseire, M., and Choong, Y. W. Mining multidimensional and multilevel sequential patterns. *ACM Trans. Knowl. Discov. Data 4*, 1 (2010), 4:1–4:37.

[91] Poulis, G., Skiadopoulos, S., Loukides, G., and Gkoulalas-Divanis, A. Distance-based kˆm-anonymization of trajectory data. In *IEEE 14th International Conference on Mobile Data Management (MDM)* (2013), pp. 57–62.

[92] Poulis, G., Skiadopoulos, S., Loukides, G., and Gkoulalas-Divanis, A. Select-organize-anonymize: A framework for trajectory data anonymization. In *13th IEEE International Conference on Data Mining Workshops* (2013), pp. 867–874.

[93] PriceWaterhouseCoopers. Transforming healthcare through secondary use of health data. `http://www.pwc.com/us/en/healthcare/publications/secondary-health-data.jhtml`, 2009.

[94] Reiss, S. P. Practical data-swapping: the first steps. *ACM Transactions on Database Systems 9* (March 1984), 20–37.

[95] Reiss, S. P., Post, M. J., and Dalenius, T. Non-reversible privacy transformations. In *Proceedings of the 1st ACM SIGACT-SIGMOD symposium on Principles of database systems* (New York, NY, USA, 1982), PODS '82, ACM, pp. 139–146.

[96] Robeznieks, A. Privacy fear factor arises. Modern Healthcare, 2005.

[97] Rubner, Y., Tomasi, C., and Guibas, L. J. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision 40*, 2 (2000), 99–121.

[98] Safran, C., Bloomrosen, M., Hammond, W. E., Labkoff, S., Markel-Fox, S., Tang, P., and Detmer, D. E. Toward a national framework for the secondary use of health data: An american medical informatics association white paper. *Journal of the American Medical Informatics Association 14*, 1 (2006), 1–9.

[99] Samarati, P. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering* (2001).

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

[101] Samarati, P., and Sweeney, L. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. *Technical Report, Computer Science Laboratory, SRI International*, 1998.

[102] Sehatkar, M. Embedding anonymity protection in a concept lattice based association rule mining algorithm. In *The Third Workshop for Women in Machine Learning (WiML)* (2008).

[103] Sehatkar, M. Privacy preserving publication of longitudinal health data. In *Canadian Conference on AI* (2010), pp. 412–413.

[104] Sehatkar, M., and Matwin, S. HALT: Hybrid anonymization of longitudinal transactions. In *Privacy, Security, Trust (PST)* (2013), pp. 127–134.

[105] Sehatkar, M., and Matwin, S. Clustering-based multidimensional sequence data anonymization. In *International Workshop onPrivacyand Anonymity in the Information Society (PAIS)* (2014), pp. 385–389.

[106] Sherkat, R., Li, J., and Mamoulis, N. Efficient time-stamped event sequence anonymization. *ACM Transactions on the Web (TWEB) 8*, 1 (2013), 4:1–4:53.

[107] Sloan, F. A., and Hsieh, C.-R. *Health Economics*. The MIT Press, Cambridge, MA, USA, 2012.

[108] Sweeney, L. Datafly: A system for providing anonymity in medical data. In *Proceedings of the IFIP TC11 WG11.3 Eleventh International Conference on Database Securty XI: Status and Prospects* (1998), pp. 356–381.

[109] Sweeney, L. Uniqueness of simple demographics in the u.s. population. *LIDAP-WP4 Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA*, 2000.

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

[111] Sweeney, L. *k*-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10*, 5 (2002), 557–570.

[112] Syvajarvi, A., and Stenvall, J. *Data Mining in Public and Private Sectors: Organizational and Government Applications.* Hershey, PA: IGI Global, 2010.

[113] Tamersoy, A., Loukides, G., Nergiz, M. E., Saygin, Y., and Malin, B. Anonymization of longitudinal electronic medical records. *IEEE Transactions on Information Technology in Biomedicine 16*, 3 (2012), 413–423.

[114] Tan, P.-N., Steinbach, M., and Kumar, V. *Introduction to Data Mining, (First Edition).* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[115] Tan, P.-N., Steinbach, M., and Kumar, V. *Introduction to Data Mining, (First Edition).* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[116] Tan, V. Y., and Ng, S.-K. Generic probability density function reconstruction for randomization in privacy-preserving data mining. In *Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern Recognition* (2007), MLDM '07, Springer-Verlag, pp. 76–90.

[117] Terrovitis, M., and Mamoulis, N. Privacy preservation in the publication of trajectories. In *Proceedings of the The Ninth International Conference on Mobile Data Management* (2008), MDM '08, pp. 65–72.

[118] Terrovitis, M., Mamoulis, N., and Kalnis, P. Privacy-preserving anonymization of set-valued data. In *Proceedings of VLDB* (2008), vol. 1, pp. 115–125.

[119] Truta, T. M., and Vinay, B. Privacy protection: p-sensitive k-anonymity property. In *Proceedings of the 22nd International Conference on Data Engineering Workshops* (2006), p. 94.

[120] Wang, K., Fung, B., and Yu, P. Template-based privacy preservation in classification problems. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining* (2005), pp. 466–473.

[121] Wang, K., Fung, B., and Yu, P. Handicapping attacker's confidence: an alternative to k-anonymization. *Knowledge and Information Systems 11*, 3 (2007), 345–368.

[122] Wang, K., Yu, P. S., and Chakraborty, S. Bottom-up generalization: A data mining solution to privacy protection. In *Proceedings of the Fourth IEEE International Conference on Data Mining* (2004), ICDM '04, pp. 249–256.

[123] Willenborg, L., and de Waal, T. *Elements of Statistical Disclosure Control*, vol. 155 of *Lecture Notes in Statistics*. New York: Springer, 2001.

[124] Willison, D., Emerson, C., Szala-Meneok, K., Gibson, E., Schwartz, L., and Weisbaum, K. Access to medical records for research purposes: Varying perceptions across research ethics boards. *Journal of Medical Ethics 34* (2008), 308–314.

[125] Winkler, W. Using simulated annealing for k-anonymity. *Technical Report 7, U.S. Census Bureau*, 2002.

[126] Wong, R., Li, J., Fu, A., and Wang, K. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *In ACM SIGKDD* (2006), pp. 754–759.

[127] Wong, R. C.-W., and Fu, A. W.-C. *Privacy-Preserving Data Publishing: An Overview*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.

[128] Wong, R. C.-W., Fu, A. W.-C., Wang, K., and Pei, J. Minimality attack in privacy preserving data publishing. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases* (2007), pp. 543–554.

[129] Xiao, X., and Tao, Y. Anatomy: simple and effective privacy preservation. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases* (2006), pp. 139–150.

[130] Xiao, X., and Tao, Y. Personalized privacy preservation. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data* (2006), pp. 229–240.

[131] Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., and Fu, A. W.-C. Utility-based anonymization using local recoding. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2006), KDD '06, ACM, pp. 785–790.

[132] Xu, Y., Fung, B., Wang, K., Fu, A., and Pei, J. Publishing sensitive transactions for itemset utility. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining* (2008), pp. 1109–1114.

[133] Xu, Y., Wang, K., Fu, A., and Yu, P. Anonymizing transaction databases for publication. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), pp. 767–775.

[134] Yarovoy, R., Bonchi, F., Lakshmanan, L. V. S., and Wang, W. H. Anonymizing moving objects: How to hide a mob in a crowd? In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology* (2009), EDBT '09, ACM, pp. 72–83.

[135] Zaki, M. J. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning 42*, 1-2 (2001), 31–60.

[136] Zaki, M. J., and jui Hsiao, C. Charm: An efficient algorithm for closed itemset mining. In *Proceedings of the SIAM international conference on data mining (SDM02)* (2002), pp. 457–473.

[137] Zhang, Q., Koudas, N., Srivastava, D., and Yu, T. Aggregate query answering on anonymized tables. In *ICDE 2007: Proceedings of the 23rd International Conference on Data Engineering* (2007), pp. 116–125.