The Pennsylvania State University

The Graduate School

College of Engineering

FUSION-AWARE PRIVACY AND WAREHOUSING FOR HEALTHCARE DATABASES

A Dissertation in

Computer Science and Engineering

by

Srivatsava Ranjit Ganta

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2009

The dissertation of Srivatsava Ranjit Ganta was reviewed and approved[1] by the following:

Raj Acharya
Professor of Computer Science and Engineering
Head of the Department of Computer Science and Engineering
Dissertation Adviser
Chair of Committee

Adam Smith
Assistant Professor of Computer Science and Engineering
Dissertation Co-Adviser

Ganapati Patil
Professor of Statistics

Robert Collins
Associate Professor of Computer Science and Engineering

[1]Signatures are on file in the Graduate School.

Abstract

In the current information era, data is being generated at an alarming rate. The healthcare domain is no exception. Healthcare organizations collect and maintain data from different stages of care provided to each and every patient. This starts with the collection of general demographics and disease history at patient check-in, followed by clinical and laboratory information during treatment, and finally follow-up and medical histories. In addition to patient data, huge amounts of medical literature and genome-wide data such as DNA sequences etc. are maintained. Managing this enormous information content is a daunting task for healthcare organizations. In this dissertation, we explore two key challenges faced in healthcare data management: 1. Data Privacy, and 2. Data Warehousing.

In the first and primary part, we take up the problem of data privacy. Data collected by healthcare organizations consists of sensitive information such as disease diagnosis etc. which should not be used or made available for non-medical purposes. However, such data needs to be disseminated and distributed by healthcare organizations to promote research, disease studies etc. While the dissemination and distribution of this information is beneficial, patient privacy is of foremost concern. Hence, privacy preserving dissemination of information becomes an important problem. In this dissertation, we focus on privacy preserving mechanisms for two specific dissemination scenarios: a) Data publishing, and b) Data sharing. In data publishing, we consider the challenge posed by what has come to be known as *auxiliary information* in the research community. The problem occurs when sensitive data is published in an anonymized version, and a potential adversary uses this version of data to collect auxiliary information from

other sources and then infers hidden sensitive information. Our contribution to this problem scenario is to introduce a new class of attacks based on auxiliary information called the *information fusion based privacy attacks*, where, an adversary fuses auxiliary information gathered with published anonymized releases to cause a privacy breach. We model two instances of such attacks: 1. Independent Release Based Attack, and 2. Web Based Attack. Our investigation on the effects of these attacks proves that a large class of existing solutions are indeed vulnerable in such scenarios. On the other hand, in the data sharing scenario, we consider the problem of privacy policy regulated data sharing among healthcare organizations. The problem here is that organizations may need to share sensitive data while following potentially conflicting privacy policies. We identify the properties of this problem with respect to the current healthcare system and design a solution based on the novel idea of sticky privacy policies.

In the second part, we take up the problem of data warehousing. Healthcare data is distributed among multiple organizational entities such as hospitals, clinical labs, research centers and government agencies that are controlled independently. This scenario leads to islands of data and hinders the availability of valuable global information to researchers. Furthermore, data mining and knowledge discovery on these islands of data leads to limited results as the data does not capture intrinsic relationships among inter-related sources. This brings out the need for data warehouse platforms that offer single-point access to patient, clinical, and genomic data from multiple sources and fusion based knowledge discovery tools that mine multiple inter-related data sets in an integrated manner. In the final part of this dissertation, we present one such system, FUZEBASE, that delivers these functionalities for cancer research data as part of a consortium of cancer centers in the state of Pennsylvania.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I am deeply grateful to my adviser Dr. Raj Acharya, for giving me the opportunity to take up grad school and pursue research. His support, guidance and patience have helped me carve myself not only as a researcher but also as a person. I am also grateful to my co-adviser Dr. Adam Smith for his guidance and mentoring. His constant search for technical excellence was a source of inspiration during my research. I would also like to thank my thesis committee members Dr. Ganapati Patil and Dr. Robert Collins for their valuable time and effort. I have enjoyed the company of many friends during my years at Penn State. Special mention goes to Silpa, Srinivas and Aravindhan. Silpa has been a source of heartiest support and has been there for me everytime I needed. Srinivas and Aravindhan have been invaluable friends to me; Srinivas, during my undergraduate days and Aravindhan, through my graduate studies. Other friends include Aakrosh, Amitayu, Angshuman, Hari, Hari Krishna, Lav, Shariff and Shiva Chaitanya. Finally, this thesis, and my life would not have been possible but for my family. My father Prabhakara Rao Ganta instilled in me the desire to learn and excel in my endeavors. My mother Dhanalakshmi Kumari Ganta has been an endless source of love and care for me. My sister Ashakiran Ganta has been a source of all that I could ask for from a delightful sibling. The two other people who love me the most are my maternal grandparents. My grandfather Krishnamurthy Dovari through his life has taught me that with self-belief and patience one can achieve anything. He remains as one of the most inspiring individuals in my life. My grandmother Vajramma Dovari constantly amazes me with her unassuming love, affection and service that she renders on our family.

Dedicated to my parents Prabhakara Rao Ganta and Dhanalakshmi Kumari Ganta and my maternal grandparents Krishnamurthy Dovari and Vajramma Dovari.

Chapter 1

Introduction

In the current information era, data is being generated at an alarming rate. The healthcare domain is no exception. Healthcare organizations collect and maintain data from each and every patient they treat. The data collected comes from all the distinct stages of care provided to the patient. This starts with general demographics, disease history and vitals at patient check-in, followed by, diagnosis, clinical and laboratory information during treatment, and finally, follow-up and medical histories. In addition to patient data, healthcare organizations maintain other information such as DNA sequences, protein sequences etc. Managing this enormous information content is a daunting task for healthcare organizations. In this dissertation, we explore two key challenges faced in healthcare data management: 1. Data Privacy, and 2. Data Warehousing.

1.1 Data Privacy

Privacy is one of foremost concerns brought about by the information era. Sensitive personal information about individuals is collected, stored and analyzed in the course of everyday life. In the United States, at least three independent credit reporting agencies maintain databases of personal information about millions of people and are widely used for credit evaluation. This includes conviction histories, mortgage histories and sometimes even phone numbers. Supermarkets and other retailers maintain and analyze large databases of customer purchase information.

Numerous websites and service providers track users search requests and navigation patterns to provide targeted services. Even though these courses of sensitive data collection seem harmless on the surface, they can easily fall into adversarial hands leading to a privacy breach. An instance of this was demonstrated in the summer of 2006 when AOL distributed search histories for more than half a million of it's users (with names removed). Nonetheless, the New York Times was able to identify a handful of users based on the content of their searches Barbaro & Zeller (2006). More recently, data released by Netflix has shown to reveal considerable amount of sensitive information Narayanan & Shmatikov (2006). Given the ease with which such data is collected and distributed, it is not surprising that many questions have been raised in recent years about individual privacy in the digital world. Broadly speaking, the problem of data privacy encompasses the many legal, ethical, and technical issues surrounding data ownership, collection, dissemination, and use.

In the healthcare domain, data privacy is a more serious concern because of the sensitivity of the data. Recently, the US Department of Health and Human Services announced a major initiative toward digitizing the patient records maintained by hospitals, pharmacies, etc. Data maintained by healthcare organizations consists of sensitive information such as disease diagnosis etc. which should not be used or made available for non-medical purposes. In view of this, the Health Insurance Portability and Accountability Act of 1996 (HIPAA) (U.S. Department of Health and Human Services Office for Civil Rights. HIPAA administrative simplification regulation text (2006)) was proposed to safeguard individual privacy. In response to this legislation, the U.S. Department of Health and Human Services issued the regulation "Standards for Privacy of Individually Identifiable Health Information", commonly known as the HIPAA Privacy Rule.

1.1.1 HIPAA

The HIPAA Privacy Rule provides two distinct sets of requirements for *de-identifying* data. By satisfying one of these two provisions, data may be exempt from many of the regulations concerning personally-identifiable health information. The first provision is deliberately vague, stating that a covered entity may determine that health information is not individually identifiable if:

A person with appropriate knowledge of, and experience with, generally accepted statistical and scientific principles and methods for rendering information not individually identifiable:

1. *Applying such principles and methods, determines that the risk is very small that the information could be used, alone or in combination with other reasonably available information, by an anticipated recipient to identify an individual who is a subject of the information.*

2. *Documents the methods and results of the analysis that justify such determination.*

In contrast, the second provision (the so-called Safe Harbor) is very specific, and quite restrictive, requiring that eighteen specific types of information, including names and geographic information, be removed entirely for any person (e.g., patients, doctors, etc.) before the data can be considered de-identified (U.S. Department of Health and Human Services Office for Civil Rights. HIPAA administrative simplification regulation text (2006)). This information is provided in the Appendix. From a technical perspective, neither of these provisions is entirely satisfactory. The first provision is not explicit about what information is considered sensitive, what constitutes a low risk, or who should be a statistical expert. The second provision is more

precise, but necessitates removing much of the information that is most useful for public health studies (e.g., geography and dates).

1.1.2 Privacy Preserving Data Dissemination

The HIPAA regulation makes privacy preservation a key challenge in healthcare data collection and management. This could be achieved in a trivial way by increasing the level of restriction on access to sensitive data collected. However, to make real use of the data collected, healthcare organizations need to share such data with third parties such as research centers, labs etc. This sharing helps promote research, drug discovery, disease studies and other benefits. Hence, privacy preserving dissemination of data becomes an important problem.

On a more generic note, sensitive data collection is carried out in different ways through different methodologies depending on the specific scenario. These can be broadly classified based on whether or not the data collector is assumed to be trusted Machanavajjhala et al. (2006).

- **Trusted Data Collector:** In this scenario, individuals providing the data trust the data collector not to breach their privacy. Examples of such data collectors are the Census Bureau, hospitals, health insurance providers, etc. So, true data is provided by the individuals to the data collector under the assumption that data privacy is ensured. To disseminate the data thus collected in a privacy preserving manner, data collectors use various methodologies:

    1. Privacy Preserving Data Publishing : In this model, the data collector releases a modified version of the data publicly making sure no sensitive information is disclosed.

2. Privacy Preserving Data Sharing : In this model, the data collector shares the data in its true form without any modifications, but only with trusted parties (e.g., other hospitals, research centers etc.).

3. Private Collaborative Computation: In this model, the data collector does not share the data, but provides support for collaborative computations which disclose no information beyond the final answer.

4. Statistical Databases: In this model, the data collector hosts a query answering service and employs techniques such as query restriction, query auditing, data perturbation and output perturbation to guarantee privacy.

- **Untrusted Data Collector:** In this scenario, the data collector is not trusted, and the private information of the individuals should be kept secret from the data collector itself. In this case, individuals themselves or the collection system provide randomized versions of their data to the data collector who then uses it for data mining. The data can then be disseminated by the data collector without any restrictions.

In the healthcare domain, patients trust the care provider in maintaining data privacy and provide a true account of their personal information. Hence, in this dissertation, we focus on trusted data collection. We further sharpen our focus on the first two dissemination methodologies used by trusted data collectors: 1.Privacy Preserving Data Publishing, and 2. Privacy Preserving Data Sharing. In data publishing, we consider the challenge posed by what has come to be known as *auxiliary information* in the research community. The problem occurs when sensitive data is published in an anonymized version and a potential adversary uses this to collect auxiliary information from other sources and then infers hidden sensitive information. Our

contribution to this problem scenario is to introduce a new class of attacks based on auxiliary information called the *information fusion based privacy attacks*, where, an adversary fuses auxiliary information gathered with published anonymized releases to cause a privacy breach. We model two instances of such attacks: 1. Independent Release Based Attack, and 2. Web Based Attack. Our investigation on the effects of these attacks prove that a large class of existing solutions are indeed vulnerable in such scenarios. On the other hand, in the data sharing scenario, we consider policy regulated data sharing among healthcare organizations. The problem here is that organizations may need to share sensitive data while following potentially conflicting privacy policies. We identify the properties of this problem with respect to the current healthcare system and design a solution based on the novel idea of sticky privacy policies.

1.2 Datawarehousing

Apart from data privacy, two aspects that carry considerable significance in healthcare data management domain are *Data Availability* and *Data Mining*. As mentioned earlier, healthcare data composes of a variety of information such as patient data, clinical data, genomic data such as DNA sequences and gene expressions etc. However, these data sets are managed by disparate organizations: hospitals, clinical labs, research centers and government agencies that are controlled independently. Hence, researchers typically operate only with islands of data. This scenario poses a serious challenge for a global study of the disease and brings out the need for platforms that offer single-point access to patient, clinical, and genomic data from multiple sources. Data warehousing encompasses the architectures, algorithms and tools for bringing together selected data from multiple databases and information sources. Traditionally data access

in such scenarios is achieved by the *lazy* or *on-demand* approach. This involves a two step process: 1. Accept a query, determine the appropriate set of information sources to answer the query and then fire the sub-queries to corresponding data sources. 2. Retrieve the results back from each repository and compute the final answer for the user. The disadvantage of this approach is that data is not retrieved until a query is fired and involves some delay. Datawarehousing involves the alternative approach of prefetching the data so that specific analysis queries can be answered in an optimized way. This approach yields better results than the *on-demand* approach when the system is targeted at specific data analysis and exploration operations.

Furthermore, data mining and knowledge discovery on these datasets leads to progress in disease diagnosis, treatment and drug discovery. However, due of the lack of data availability it is typically carried out exclusively on each of the datasets. This leads to limited knowledge discovery as the data does not capture the intrinsic relationships between these inter-related information sources. For example, biomarker studies are one of the key knowledge discovery tasks involved in disease research that identifies genes that are responsible for the disease. This is traditionally achieved by running cluster analysis on gene expression data alone. Recent advancement in this area Holmes & Bruno (2000) Kasturi & Acharya (2004) shows that inclusion of clinical and other related data in such studies leads to better results.

These observations bring out the need for platforms that offer tools to mine and explore across heterogeneous healthcare data. In the final part of this dissertation, we present, FUZEBASE, a system that delivers these functionalities for cancer research data. This system was developed as part of a consortium of cancer centers in the state of Pennsylvania.

1.3 Outline

The rest of this dissertation is organized as follows:

Chapter 2 focuses on privacy preserving data publishing. Provides a brief introduction to data publishing and discusses prior work done in this area. Identifies the current challenges in the problem domain in the form of information fusion based privacy attacks on existing solutions. Introduces the two adversarial attacks modeled: 1. Independent Release Based Attack, and 2. Web Based Attack.

Chapter 3 investigates the independent release based attack. Presents the attack setting and simulates the attack on real data. Provides an in-depth experimental study on the extent of breach possible and reason about the properties of current solutions that lead to the success of such attacks.

Chapter 4 investigates the web based attack. Presents the attack setting and simulates the attack on real data. Provides an experimental study on the feasibility of the attack and presents a solution strategy.

Chapter 5 shifts focus to privacy preserving data sharing. Provides a brief introduction to data publishing and discusses prior work done in this area. Introduces the problem of privacy preserving cross-enterprise data sharing.

Chapter 6 investigates the cross-enterprise data sharing problem. Proposes a solution by introducing sticky privacy policies and presents a system implementation of the solution.

Chapter 7 motivates the need for data warehousing and information fusion in healthcare and presents a data warehouse system and fusion platform built as part of a cancer consortium.

Chapter 8 provides the conclusions and future directions.

1.4 Contributions

The contributions made by this research to various branches of research communities can be summarized as follows:

Contribution to Computer Science:

1. The research carried out in this dissertation contributes to two sections of computer science research community: 1. Privacy 2. Medical Informatics.
2. The work done touches upon issues faced by several sections of audiences in the healthcare domain including administrators, policy designers and patients.
3. The outcomes will contribute to better protection of individual privacy rights while promoting the sharing and dissemination of sensitive data.

Contribution to Privacy Research:

1. The research contributes to the identification and study of information fusion based privacy attacks.

2. The outcomes help develop better anonymization techniques and stronger guarantees on privacy protection in case of such attacks.

3. The work done helps create a stronger and more generic definition of anonymity.

Contribution to Biomedical Informatics Research:

1. The work done serves the medical informatics research community by providing an access platform for cancer related data.

2. The outcomes can be readily used as an experimentation and data mining platform for cancer research.

Chapter 2

Privacy Preserving Publishing

In the previous chapter, we discussed how privacy preserving publishing and sharing are an important class of problems in healthcare data management. In this chapter we expand upon the first part, *privacy preserving publishing*.

Data publishing arises in the setting where a central organization, such as a hospital, manages large collection of individual-specific data. Where, the data consists of sensitive records such as patient discharge information, electronic medical records etc with one or more records per individual. The organization intends to release this data to the public for purposes such as research, dissemination etc which is often required by law. However, such a release should achieve two goals:

- The data published should not compromise the *privacy* of the individuals in the database.
- The data published should carry enough *utility* for certain intended tasks (data mining etc.) to be run.

This setting is depicted in Figure 2.1. So, the goal in privacy preserving data publishing is to transform the data such that global statistics of the data (utility) are released without compromising individual privacy.

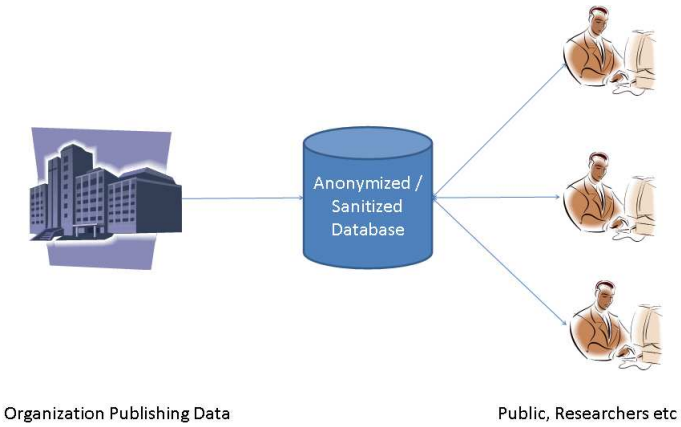


Fig. 2.1: Data Publishing

2.1 Background and Related Work

Work on data publishing originated from the Census Bureau where the goal was to release census data as public-use microdata (PUMS). A variety of *sanitization* techniques were used to ensure privacy and utility in microdata. As a result, there was a huge amount of research on data sanitization in the statistics community. A survey of the work done is made by Adam & Wortmann (1989). Census data literature focuses on identifying and protecting the privacy of sensitive entries in contingency tables - tables of counts which represent the complete cross-classification of the data I.P.Fellegi (1972) Cox (1980) Cox (1982) Cox (1987) Dobra & Feinberg (2003) Dobra & Feinberg (2000) Slavkovic & Feinberg (2004). A non-zero table entry is considered sensitive if it is smaller than a fixed threshold which is usually chosen in an ad-hoc manner. Two main approaches have been proposed for protecting the privacy of sensitive cells: *data swapping* and *data suppression*. The data swapping approach involves moving data entries from one cell in the contingency table to another so that the table remains consistent with a set of published marginals Dalenius & Reiss (1982) Duncan & Feinberg (1997). In the data suppression approach Cox (1980) Cox (1995), cells with low counts are simply deleted. Due to data

dependencies caused by marginal totals that may have been previously published, additional related cell counts may also need to be suppressed. An alternate approach is to determine a safety range or protection interval for each cell Dobra (2002), and to publish only those marginals which ensure that the feasibility intervals (i.e. upper and lower bounds on the values a cell may take) contain the protection intervals for all cell entries.

Computer science research has also tried to solve the privacy preserving data publishing problem. Sweeney (2002) shows that publishing data sets for which the identifying attributes (keys) have been removed is not safe and may result in privacy breaches. In fact, the paper shows a real life privacy breach using health insurance records and voter registration data. To better protect the data, Sweeney (2002) proposes the use of k-anonymity Samarati & Sweeney (1998) which ensures that every individual is hidden in a group of size at least k with respect to the non-sensitive attributes. The problem of k-anonymization is NP-hard Meyerson & Williams (2004a); approximation algorithms for producing k-anonymous tables have been proposed Aggarwal et al. (2004). Chawla et al. (2005) proposes a formal definition of privacy for published data based on the notion of blending in a crowd. Here, privacy of an individual is said to be protected if an adversary cannot isolate a record having attributes similar (according to a suitably chosen distance metric) to those of a given individual without being sufficiently close (according to the distance metric) to several other individuals; these other individuals are the crowd. The authors propose several perturbation and histogram-based techniques for data sanitization prior to publication. The formalization of the notion of privacy presents a theoretical framework for studying the privacy-utility trade-offs of the proposed data sanitization techniques. However, due to the heavy reliance on an inter-tuple distance measure of privacy, the proposed definition of privacy fails to capture scenarios where identification of even a single sensitive attribute may

constitute a privacy breach. Miklau & Suciu (2004) characterize the set of views that can be published while keeping some query answer secret. Privacy here is defined in the information-theoretic sense of perfect privacy. They show that to ensure perfect privacy, the views that are published should not be related to the data used to compute the secret query. This shows that perfect privacy is too strict as most useful views, like those involving aggregation, are disallowed. Finally, there has been some work on publishing XML documents and ensuring access control on these documents Miklau & Suciu (2003) Yang & Li (2004). Miklau & Suciu (2003) use cryptographic techniques to ensure that only authorized users can access the published document. Yang & Li (2004) propose publishing partial documents which hide sensitive data. The challenge here is that the adversary might have background knowledge which induces dependencies between branches, and this needs to be taken into account while deciding which partial document to publish.

2.1.1 Data Anonymization

In this dissertation, we focus primarily on work done in the computer science community. We roughly classify this work into two broad classes:

- ***Randomization* based anonymization schemes :** Schemes that introduce uncertainty either by randomly perturbing the raw data (a technique called *input perturbation*, *randomized response*, e.g., Warner (1965); Agrawal & Srikant (2000); Evfimievski et al. (2002)), or *post-randomization*, e.g., van den Hout & van der Heijden (2002)), or by injecting randomness into the algorithm that is used to analyze the data (e.g., Blum et al. (2005); McSherry & Talwar (2007)).

- ***Partitioning* based anonymization schemes :** Schemes that guarantee privacy by partitioning the data such that an adversary cannot uniquely identify the individuals falling in each partition. The basic ideology behind these techniques is *blending in the crowd* which guarantees that an individual or entity cannot be distinguished from a minimum number of other people. So, they partition individuals in the database into disjoint groups satisfying certain criteria (for example, in k-anonymity Sweeney (2002), each group must have size at least k). For each group, certain exact statistics are calculated and published. Partition-based schemes include k-anonymity Sweeney (2002) as well as several recent variants, e.g., Machanavajjhala et al. (2007); Xiao & Tao (2006a,b); Li et al. (2007); Xiao & Tao (2007); Martin et al. (2007); Chen et al. (2007); LeFevre et al. (2006a).

We now further elaborate on partitioning based anonymization schemes, as our work is directly related to this particular class of solutions.

2.1.1.1 Partitioning based Schemes

The basic model followed by partition based schemes assumes that individual-specific data is stored in a table (or a relation) of columns (or attributes) and rows (or records). Table 2.1 depicts a typical database. Each data attribute is uniquely categorized into of the three different types based on application domain knowledge:

1. **Identifier Attributes:** Attributes carrying explicit identifiers such as *Name, SSN* etc.

2. **Quasi Identifier Attributes:** Attributes that could indirectly lead to identification of individuals in the database such as *Age*, *Zipcode* and *Gender* etc. These are also sometimes referred to as *Non-Sensitive* attributes.

3. **Sensitive Attributes:** Attributes carrying the sensitive information about the individuals such as *Disease*, *Income* etc.

Identifiers		**Quasi Identifiers**			**Sensitive**
Name	**SSN**	**Zipcode**	**Age**	**Nationality**	**Condition**
Alice	111-111-1111	13053	28	Russian	AIDS
Bob	222-222-2222	13068	29	American	Flu
Christine	333-333-3333	13068	21	Japanese	Cancer
Robert	444-444-4444	13053	23	American	Meningitis

Table 2.1: Typical Sensitive Database in Partitioning Based Schemes

The anonymized version of the database contains only non-sensitive and sensitive attributes with the identifier attributes completely removed. The non-sensitive attributes are transformed into a form such that each record is indistinguishable from certain other records while the sensitive attributes are published with actual values.

2.1.1.2 k-anonymity

k-anonymity is the first and most popular partitioning based anonymization technique proposed. It originated from a study which estimated that 87% of the population of the United States could be uniquely identified using the seemingly innocuous attributes of gender, date of birth, and 5-digit zip code Sweeney (2000). In fact, these three attributes were used to link Massachusetts voter registration records (which included the name, gender, zip code, and date of birth) to supposedly anonymized medical data from GIC1 (which included gender, zip code, date of birth and diagnosis). This linking attack managed to uniquely identify the medical records of the governor of Massachusetts in the medical data Sweeney (2002). Sets of attributes (like gender, date of birth, and zip code in the previous example) that can be linked with external data

to uniquely identify individuals in the population are known to be called as quasi-identifiers. To counter linking attacks using quasi-identifiers, Samarati and Sweeney proposed the definition of k-anonymity Sweeney (2002).

A table/database satisfies k-anonymity if every record in the table is indistinguishable from at least $k-1$ other records with respect to every set of quasi-identifier attributes. Hence, for every combination of values of the quasi-identifiers in the k-anonymous table, there are at least k records that share those values. This ensures that individuals cannot be uniquely identified by linking attacks.

Example. Table 2.2 shows medical records from a fictitious hospital. Note that the table contains no uniquely identifying attributes like name, social security number, etc. In this example, we divide the attributes into two groups: the sensitive attributes (consisting only of medical condition) and the nonsensitive attributes (zip code, age, and nationality). An attribute is marked sensitive if an adversary must not be allowed to discover the value of that attribute for any individual in the dataset. Attributes not marked sensitive are nonsensitive. Furthermore, let the collection of attributes zip code, age, nationality be the quasi-identifiers for this dataset. Tables 2.3 shows a 4-anonymous version of Table 2.2 (here "*" denotes a suppressed value so, e.g., "zip code = 1485*" means that the zip code is in the range $[1485014859]$ and "age = 3*" means the age is in the range $[3039]$). Note that in the 4-anonymous table, each tuple has the same values for the quasi-identifier as at least three other tuples in the table.

Because of its conceptual simplicity, k-anonymity has been widely discussed as a viable definition of privacy in data publishing.

Generalization and *Suppression* were the two techniques initially employed to achieve k-anonymity. Generalization involves replacing (or recoding) a value with a less specific but

	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 2.2: Sample Patient Data

	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*	$\geq$40	*	Cancer
6	1485*	$\geq$40	*	Heart Disease
7	1485*	$\geq$40	*	Viral Infection
8	1485*	$\geq$40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Table 2.3: 4-anonymous version of Patient Data

semantically consistent value. Samarati proposes a generalization based framework to achieve k-anonymity Samarati & Sweeney (1998) and provides a binary search algorithm for discovering a single minimal generalization. This transformation is made at the cost of information lost during the anonymization and is minimized using an optimization function to attain a specific level of k-anonymity. Various measures and optimization functions have been proposed Iyengar (2002) to deal with the minimization of information loss resulting from k-anonymization. Given such a cost metric, genetic algorithms Iyengar (2002) and simulated annealing Winkler (2002) have been considered for finding locally minimal anonymizations, using the single-dimension full sub-tree recoding model for categorical attributes and the single-dimension ordered-set partitioning model for numeric data. Recently, top-down Fung et al. (2005) and bottom-up Wang et al. (2004) greedy heuristic algorithms were proposed for producing anonymous data that remains useful for building decision-tree classifiers. Bayardo & Aggarwal (2005) propose a top-down set-enumeration approach for finding an anonymization that is optimal according given the single-dimension ordered-set partitioning model. Subsequent work shows that optimal anonymizations under this model may not be as good as anonymizations produced with a multi-dimension variation LeFevre et al. (2006a). The minimal cell and attribute-suppression variations of k-anonymization were shown to be NP-hard (with hardness proofs constructed based on the number of cells and number of attributes, respectively), and $O(klogk)$ Meyerson & Williams (2004b) and $O(k)$ Aggarwal et al. (2005) approximation algorithms were proposed. Finally, a generalization based anonymization was proposed in Xiao & Tao (2006b) based on personalized preferences.

Microaggregation is a family of k-anonymization techniques for *quantitative (numeric)* data. The rationale behind microaggregation is that, for a dataset with n data vectors, the data

vectors are clustered to form g groups each of size at least k. For each variable, the average value over each group is computed and is used to replace the original data vector. The groups are formed such that the information loss due to anonymization is minimal. The loss in the information is measured based on the within-group homogeneity the higher the within-group homogeneity, the lesser is the information loss. The homogeneity or similarity of the group is measured by the well-known sum-of-squares distance from the centroid. An optimal k-partition is thus defined to be the one with minimal sum-of-squares. Note that the partition problem implicit in microaggregation differs from the classical clustering problem whose goal is to split a population into a fixed number of disjoint groups, regardless of the group size. In microaggregation, the constraint is on the group size rather than on the number of groups. The problem of optimal microaggregation is related, but not identical, to the minimum sum-of-squares clustering *(MSSC)*, whose goal is to find a partitioning of a data set into fixed number of disjoint groups (without size constraint) so that the within-group sum of squares is minimum. The *MSSC* problem is known to be NP-hard Brucker (1978). To the best of our knowledge, neither complexity assessment nor polynomial exact algorithms are available in the literature for optimal microaggregation. However, several heuristic solutions such as Domingo-Ferrer & Mateo-Sanz (2002) Mateo-Sanz & J.Domingo-Ferrer (1878) have been proposed for obtaining optimal *univariate* microaggregation and *multivariate* microaggregation. In our work, we use multivariate fixed size microaggregation method proposed in Domingo-Ferrer & Mateo-Sanz (2002).

2.2 Challenges

Several challenges arise in privacy preserving data publishing. One of the most significant is *auxiliary information*, also called external knowledge, background knowledge, or side

information. Consider the scenario where an organization intends to publish sensitive data about a set of individuals. The organization uses one of the existing solutions to anonymize the data and deems it safe to publish. Now, either by using the published data or from external sources an adversary could collect auxiliary information about the individuals. Such auxiliary information can be gathered from a variety of publicly available sources such as the web, voter records, releases from other organizations etc. The adversary can then *fuze* this auxiliary information with the anonymized release to infer otherwise protected information and cause a privacy breach.

In this dissertation, we study adversarial attacks involving rich, realistic sources of auxiliary information and refer to them as *Information Fusion based Privacy Attacks*. Specifically, we model two attacks where the auxiliary information is collected from, a) Independent anonymized releases from multiple organizations about overlapping populations b) Web-based sources for individual-specific information such as personal homepages, blogs etc. The attack model is illustrated in Figure 2.2.

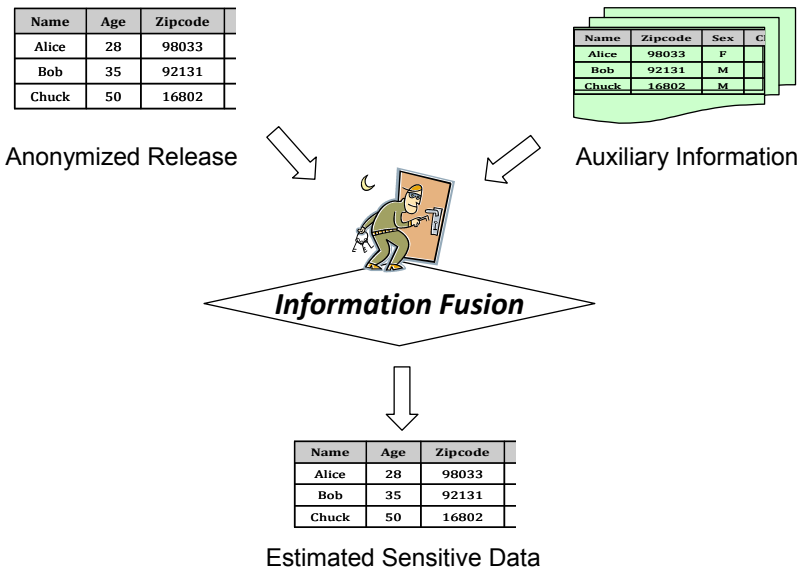


Fig. 2.2: Information Fusion based Privacy Attacks

2.2.1 Information Fusion Based Privacy Attacks

- In the first form of information fusion attack, we investigate the scenario where the adversary obtains auxiliary information from anonymized releases made by multiple organizations about overlapping populations. Consider the scenario where two hospitals located in the same city publishing anonymized patient discharge data. Because of the proximity of their location, some of the patients may visit both hospitals with similar ailments. Hence, their records would be present in both the releases. Now, say an adversary obtains quasi-identifying information about one of those individuals who visited both the hospitals. He could then put together the releases to infer otherwise protected data about the individual. We investigate this problem of Independent Release based Attack in Chapter 4.

- In the second form of information fusion attack, we investigate the scenario where the adversary obtains auxiliary information from web based sources. The present day world-wide-web carries a lot of individual-specific information through personal homepages, blogs and other community portals. The information obtained through these sources can be put together with the anonymized release through information fusion or data mining techniques to infer or predict sensitive data. We investigate this problem of Web based Attack in Chapter 5.

Chapter 3

Information Fusion Attack 1: Independent Release Based Attack

In this chapter, we investigate the problem of Independent Release based Attack. As mentioned in the previous chapter, we focus on partitioning-based anonymization schemes and our goal is to study the effects of the attack on such schemes. Section 3.1 introduces the problem. Section 3.2 discusses related work and Section 3.3 provides the definitions involved and formulates the attack. Section 3.4 demonstrates the attack on a real dataset and provides an in-depth experimental study. Section 3.5 provides the conclusions.

3.1 Problem

Let us consider a simple example to introduce independent release based attacks. Consider two hospitals H_1 and H_2, that are located in the same city and plan to publish their patient-discharge databases. Tables 3.1 and 3.3 present examples of patient discharge databases from H_1 and H_2 respectively. Using a partitioning based scheme such as k-anonymity, H_1 and H_2 anonymize this data possibly using different values of k, say $k_1 = 4$ and $k_2 = 6$ respectively. Tables 3.2 and 3.4 represent the anonymized versions of corresponding patient discharge data from H_1 and H_2.

Because they are in the same city, some patients may visit both H_1 and H_2. Assume that an adversary, possibly an employer, knows that $Alice$ is a 28 years old female living in zip code 13012 and that she recently visited both the hospitals. Using this information,

the adversary can look-up Alice in the anonymized releases from H_1 and H_2. From $H_1's$ release (Table 4.2), he concludes that an individual who is 28 years old and living in zip-code 13012 falls into the first partition. From this, he deduces that the possible sensitive attribute values for Alice are $S_1 = \{AIDS, HeartDisease, ViralInfection\}$. Similarly, from $H_2's$ release (Table 4.4), he concludes that the possible sensitive attribute values are $S_2 = \{AIDS, Tuberculosis, Flu, Cancer\}$. Since the adversary knows that Alice has visited both the hospitals, by doing a simple set intersection on S_1 and S_2, he can conclude that the only possibility of sensitive attribute value for Alice is $AIDS$.

Table 3.1: H_1 Patient Data

	Non-Sensitive			**Sensitive**
	Zipcode	**Age**	**Nationality**	**Condition**
1	13053	28	Russian	AIDS
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	13853	50	Indian	Cancer
6	13853	55	Russian	Heart Disease
7	13850	47	American	Viral Infection
8	13850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

Note that the above example relies on two properties of the partitioning-based anonymization schemes:

1. **(i) Exact sensitive value disclosure:** As explained in the previous chapter, partitioning-based schemes release the exact "sensitive" value corresponding to each member of the group.

Table 3.2: 4-anonymous H_1 Patient Data

	Non-Sensitive			**Sensitive**
	Zipcode	**Age**	**Nationality**	**Condition**
1	130**	$<$30	*	AIDS
2	130**	$<$30	*	Heart Disease
3	130**	$<$30	*	Viral Infection
4	130**	$<$30	*	Viral Infection
5	130**	$\geq$40	*	Cancer
6	130**	$\geq$40	*	Heart Disease
7	130**	$\geq$40	*	Viral Infection
8	130**	$\geq$40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Table 3.3: H_2 Patient Data

	Non-Sensitive			**Sensitive**
	Zipcode	**Age**	**Nationality**	**Condition**
1	13053	28	Russian	AIDS
2	13067	31	Philippines	Tuberculosis
3	13068	23	Russian	Flu
4	13053	24	Indian	Tuberculosis
5	13854	49	Indian	Cancer
6	13853	55	Russian	Tuberculosis
7	13850	47	American	Viral Infection
8	13850	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: 6-anonymous H_2 Patient Data

	Non-Sensitive			Sensitive
	Zipcode	**Age**	**Nationality**	**Condition**
1	130**	<35	*	AIDS
2	130**	<35	*	Tuberculosis
3	130**	<35	*	Flu
4	130**	<35	*	Tuberculosis
5	130**	<35	*	Cancer
6	130**	<35	*	Cancer
7	130**	≥35	*	Cancer
8	130**	≥35	*	Cancer
9	130**	≥35	*	Cancer
10	130**	≥35	*	Tuberculosis
11	130**	≥35	*	Viral Infection
12	130**	≥35	*	Viral Infection

2. **(ii) Locatability:** Given any individual's non-sensitive values (non-sensitive values are exactly those that are assumed to be obtainable from other, public information sources), one can locate the group in which the individual has been put in.

Based on these properties, an adversary can narrow down the set of possible sensitive values for an individual by intersecting the sets of sensitive values present in his/her groups from multiple anonymized releases. Properties (i) and (ii) turn out to be widespread. Exact sensitive value disclosure is a design feature common to all partitioning-based anonymization schemes. Locatability is less universal, since it depends on the exact choice of partitioning algorithm (used to form groups) and the recoding applied to the non-sensitive attributes. However, some schemes always satisfy locatability by virtue of their structure (e.g., schemes that recursively partition the data set along the lines of a hierarchy that is subsequently used for generalization LeFevre et al. (2006a,b), or schemes that release the exact set of non-sensitive attribute vectors for each group Xiao & Tao (2006a)). For other schemes, locatability is not perfect but our experiments suggest that using simple heuristics one can locate a individual's group with high probability.

Even with these properties, it is difficult to come up with a theoretical model for independent release based attacks because the partitioning techniques generally create dependencies that are hard to model analytically. Instead, we evaluated the success of independent release attacks empirically. We ran the independent release attack on two popular census databases anonymized using partitioning-based schemes. We evaluated the severity of the attack by measuring the number of individuals who had their sensitive value revealed. Our experimental results confirm that partitioning-based anonymization schemes including k-anonymity and its recent variants, ℓ-diversity and t-closeness, are indeed vulnerable to independent release attacks. Section 3.4 elaborates our methodology and results.

3.2 Related Work

Previous line of work on independent release based attacks focused on taking into account other, *known* releases, such as previous publications by the same organization (called "sequential" or "incremental" releases, Wang & Fung (2006); Byun et al. (2006); Xiao & Tao (2007); Fung et al. (2008); Pei et al. (2007)) and multiple views of the same data set Yao et al. (2005). However, our problem deals with the scenario where the publisher is not aware of other anonymized releases. Another line of work has considered incorporating knowledge of the partitioning algorithm used to group individuals Wong et al. (2007). Most relevant are works that have sought to model *unknown* background knowledge. Martin et al. (2007) and Chen et al. (2007) provide complexity measures for an adversary's side information (roughly, they measure the size of the smallest formula within a CNF-like class that can encode the side information). Both works design schemes that provably resist attacks based on auxiliary information whose complexity is below a given threshold.

Independent releases fall outside the models proposed by these works. The sequential release models do not fit because they assume that other releases are known to the anonymization algorithm. The complexity-based measures do not fit because independent releases appear to have complexity that is linear in the size of the data set.

3.3 Definitions

Let D be a multiset of tuples where each tuple corresponds to an individual in the database. Let R be an anonymized version of D. From this point on, we use the terms tuple and individual interchangeably, unless the context leads to ambiguity. Let $A = A_1, A_2, \ldots, A_r$ be a collection of attributes and t be a tuple in R; we use the notation $t[A]$ to denote $(t[A_1], \ldots, t[A_r])$ where each $t[A_i]$ denotes the value of attribute A_i in table R for t.

Definition 1 (Quasi-identifier) *A set of non-sensitive attributes $\{Q_1, \ldots, Q_r\}$ is called a **quasi-identifier** if there is at least one individual in the original sensitive database D who can be uniquely identified by linking these attributes with auxiliary data.*

Definition 2 (Equivalence Class) *An **equivalence class** for a table R with respect to attributes in A is the set of all tuples $t_1, t_2, \ldots, t_i \in R$ for which the projection of each tuple onto attributes in A is the same, i.e., $t_1[A] = t_2[A] \ldots = t_i[A]$.*

Definition 3 (k-anonymity Sweeney (2002)) *A release R is **k-anonymous** if for every tuple $t \in R$, there exist at least $k-1$ other tuples $t_1, t_2, \ldots, t_{k-1} \in R$ such that $t[A] = t_1[A] = \ldots = t_{k-1}[A]$ for every collection A of attributes in quasi-identifier.*

We also consider two extensions to k-anonymity, ℓ-diversity and t-closeness.

Definition 4 (Entropy ℓ-diversity Machanavajjhala et al. (2007)) *For an equivalence class E, let S denote the domain of the sensitive attributes, and $p(E, s)$ is the fraction of records in E that have sensitive value s, then E is ℓ-**diverse** if:*

$$-\sum_{s \in S} p(E, s) \log(p(E, s)) \geq \log l \, .$$

A table is ℓ-diverse if all its equivalence classes are ℓ-diverse.

Definition 5 (t-closeness Li et al. (2007)) *An equivalence class E is t-**close** if the distance between the distribution of a sensitive attribute in this class and distribution of the attribute in the whole table is no more than a threshold t. A table is t-close if all its equivalence classes are t-close.*

Definition 6 (locatability) *Let Q be the set of quasi-identifier values of an individual in the original database D. Given the k-anonymized release R of D, the locatability property allows an adversary to identify the set of tuples $\{t_1, \ldots, t_K\}$ in R (where $K \geq k$) that correspond to Q.*

Locatability does not necessarily hold for all partitioning-based schemes, since it depends on the exact choice of partitioning algorithm (used to form groups) and the recoding applied to the non-sensitive attributes. However it is widespread. Some schemes *always* satisfy locatability by virtue of their structure (*e.g.*, schemes that recursively partition the data set along the lines of a hierarchy always provide locatability if the attributes are then generalized using the same hierarchy, or if (min,max) summaries are used LeFevre et al. (2006a,b)). For other schemes, locatability is not perfect but our experiments suggest that using simple heuristics one

can locate a person's group with good probability. For example, microaggregation Domingo-Ferrer & Mateo-Sanz (2002) clusters individuals based on Euclidean distance. The vectors of non-sensitive values in each group are replaced by the centroid (i.e., average) of the vectors. The simplest heuristic for locating an individual's group is to choose the group with the closest centroid vector. In experiments on census data, this correctly located approximately 70% of individuals. In our attacks, we always assume locatability. This assumption was also made in previous studies Sweeney (2002); Martin et al. (2007).

3.3.1 Independent Release Attack

Armed with these basic definitions, we now proceed to formalize the independent release attack (Algorithm 1).

Algorithm 1 Independent Release Attack

1: $R_1, \ldots, R_n \leftarrow n$ independent anonymized releases
2: $P \leftarrow$ set of overlapping population
3: **for** each individual i in P **do**
4: **for** $j = 1$ to n **do**
5: $e_{ij} \leftarrow$ getEquivalenceClass(R_j, i)
6: $s_{ij} \leftarrow$ getSensitiveValueSet(e_{ij})
7: **end for**
8: $S_i \leftarrow s_{i1} \cap s_{i2} \cap \ldots \cap s_{in}$
9: **end for**
10: **return** $S_1, \ldots, S_{|P|}$

Let $R_1, \ldots, R_n$ be n independent anonymized releases with minimum partition-sizes of $k_1, \ldots, k_n$, respectively. Let P be the overlapping population occurring in all the releases. The function getEquivalenceClass returns the equivalence class into which an individual falls

in a given anonymized release. The function getSensitiveValueSet returns the set of (distinct) sensitive values for the members in a given equivalence class.

Definition 7 (Anonymity) *For each individual i in P, the anonymity factor proposed by each release R_j is equal to the corresponding minimum partition-size k_j.*

However, as pointed out in Machanavajjhala et al. (2007), the actual anonymity offered is less than this ideal value and is equal to the number of distinct values in each equivalence class. We call this as the *effective anonymity*.

Definition 8 (Effective Anonymity) *For an individual i in P, the effective anonymity offered by a release R_j is equal to the number of distinct sensitive values of the partition into which the individual falls. Let e_{ij} be the equivalence class or partition into which i falls with respect to the release R_j, and let s_{ij} denote the sensitive value set for e_{ij}. The effective anonymity for i with respect to the release R_j is: $EA_{ij} = |s_{ij}|$.*

For each target individual i, EA_{ij} is the *effective prior anonymity* with respect to R_j (anonymity before the independent release attack). The list of possible sensitive values is equal to the intersection of all sensitive value sets s_{ij}, $j = 1, \ldots, n$. So the *effective posterior anonymity* ($\widehat{EA}_i$) for i is:

$$\widehat{EA}_i = |\{\cap s_{ij}\}|, j = 1, \ldots, n\,.$$

The difference between the effective prior anonymity and effective posterior anonymity quantifies the drop in effective anonymity.

$$Anon\_Drop_i = \min_{j=1,\ldots,n}\{EA_{ij}\} - \widehat{EA}_i\,.$$

The *vulnerable population* (VP) is the number of individuals (among the overlapping population) for whom the independent release attack leads to a positive drop in the effective anonymity.

$$VP = \{i \in P \,:\, Anon\_Drop_i > 0\} \,.$$

After performing the sensitive value set intersection, the adversary knows only a possible set of values that each individual's sensitive attribute can take. So, the adversary deduces with equal probability (under the assumption that the adversary does not have any further auxiliary information) that the individual's actual sensitive value is one of the values in the set $\left\{\cap s_{ij}\right\}, j = 1, \ldots, n$. So, the adversaries *confidence level* for an individual i can be defined as:

Definition 9 (Confidence level C_i) *For each individual i, the confidence level C_i of the adversary in identifying the individual's true sensitive value through the independent release attack is defined as $C_i = \frac{1}{\widehat{EA}_i}$.*

Now, given some confidence level C, we denote by VP_C and PVP_C the set and the percentage of overlapping individuals for whom the adversary can deduce the sensitive attribute value with a confidence level of at least C.

$$VP_C = \{i \in P \,:\, C_i \geq C\} \,,$$

$$PVP_C = \frac{|VP_C| \cdot 100}{|P|} \,.$$

3.4 Experimental Results

3.4.1 Setup

We demonstrate the independent release attack on two census-based databases available through the UCI Machine Learning Repository (2008). The first one is the Adult database that has been used extensively in partitioning based anonymization studies. The database was prepared in a similar manner to previous studies LeFevre et al. (2006a); Machanavajjhala et al. (2007) (also explained in Table 3.5). The resulting database contained individual records corresponding to 30162 people. The second database is the IPUMS database that contains data from the 1997 census studies. We only use a subset of the attributes that are similar to attributes present in the Adult database to maintain uniformity in quasi-identifiers. The IPUMS database contains individual records corresponding to a total of 70187 people. This data set was prepared as explained in Table 3.6. From each database, we generate two overlapping subsets (Subset 1 and Subset 2) by randomly sampling individuals without replacement from the total population. We fixed the overlap size to $P = 5000$. For each database, the two subsets are anonymized independently and the independent release attack is run on the anonymization results.

We use three different partitioning-based anonymization techniques to demonstrate the attack: k-anonymity, ℓ-diversity, and t-closeness. For k-anonymity, we use the Mondrian multidimensional approach proposed in LeFevre et al. (2006a) and the microaggregation technique proposed in Domingo-Ferrer & Mateo-Sanz (2002). For ℓ-diversity and t-closeness, we use the definitions of entropy ℓ-diversity and t-closeness proposed in Machanavajjhala et al. (2007) and Li et al. (2007), respectively.

All the experiments were run on a Pentium 4 system running Windows XP with 1GB RAM.

Attribute	Domain Size	Class
Age	74	Quasi ID
Work Class	7	Quasi ID
Education	16	Quasi ID
Marital Status	7	Quasi ID
Race	5	Quasi ID
Gender	2	Quasi ID
Native Country	41	Quasi ID
Occupation	14	Sensitive

Table 3.5: Adult Census Database Description.

Attribute	Domain Size	Class
Age	100	Quasi ID
Work Class	5	Quasi ID
Education	10	Quasi ID
Marital Status	6	Quasi ID
Race	7	Quasi ID
Sex	2	Quasi ID
Birth Place	113	Quasi ID
Occupation	247	Sensitive

Table 3.6: IPUMS Census Database Description.

3.4.2 Severity of the Attack

Our first goal is to quantify the extent of damage possible through the attack. For this, we consider two possible situations: (i) Perfect breach and (ii) Partial breach.

3.4.2.1 Perfect Breach

A perfect breach occurs when the adversary can deduce the exact sensitive value of an individual. In other words, a perfect breach is when the adversary has a confidence level of 100%

about the individual's sensitive data. To estimate the probability of a perfect breach, we compute the percentage of overlapping population for whom the independent release attack leads to a final sensitive value set of size 1. Figure 3.1 ((a) and (b)) plots this result.

We consider three scenarios for anonymizing the two overlapping subsets: (i) Mondrian on both the data subsets, (ii) Microaggregation on both the data subsets, and (iii) Mondrian on the first subset and microaggregation on the second subset. The x-axis label, (k_1, k_2), represents the pair of k values used to anonymize the first and the second subset, respectively. In the experiments, we use the same k values for both the subsets $(k_1 = k_2)$. Note that for simplicity, from now on we will be defining confidence level in terms of percentages.

In the case of Adult database we found that around 12% of the population is vulnerable to a perfect breach for $k_1 = k_2 = 5$. For the IPUMS database, this value is much more severe around 60%. As the degree of anonymization increases or in other words, as the value of k increases, the percentage of vulnerable population goes down. The reason for that is that is, as the value of k increases, the partition sizes in each subset increases. This leads to a larger intersection set and thus a lower probability of obtaining an intersection set of size 1.

3.4.2.2 Partial Breach

Our next experiment aims to compute a more practical quantification of the severity of the independent release attack. In most cases, to inflict a privacy breach, all that the adversary needs to do is to boil down the possible sensitive values to a *few* values which itself could reveal a lot of information. For example, for a hospital discharge database, by boiling down the sensitive values of the disease/diagnosis to a few values, say, "Flu", "Fever", or "Cold", it could be concluded that the individual is suffering from a viral infection. In this case, the adversary's

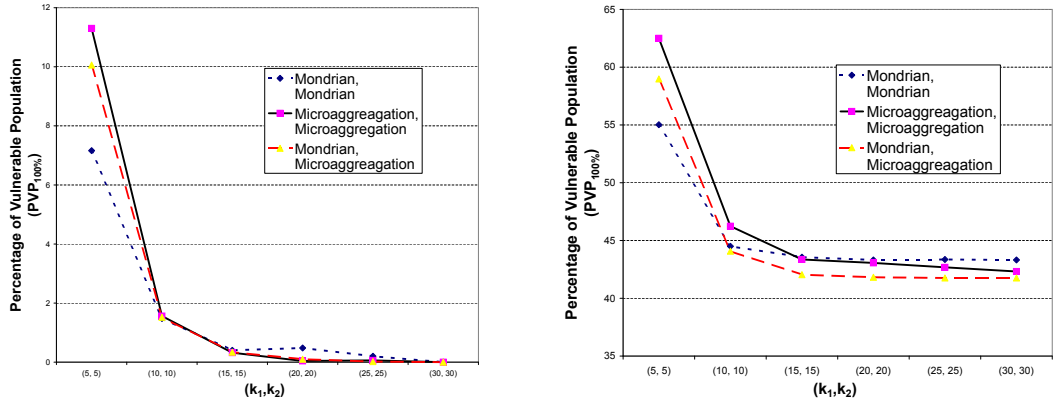


Fig. 3.1: Severity of the Independent Release Attack - Perfect Breach (a) Adult Database and (b) IPUMS Database.

confidence level is $1/3 = 33\%$. Figure 3.1 ((c) and (d)) plots the percentage of vulnerable population for whom the independent release attack leads to a partial breach for the Adult and IPUMS databases.

Here, we only use the first anonymization scenario described earlier in which both the overlapping subsets of the database are anonymized using Mondrian multidimensional technique. Observe that the severity of the attack increases alarmingly for slight relaxation of the required confidence level. For example, in the case of IPUMS database, around 95% of the population is vulnerable for a confidence level of 25% for $k_1 = k_2 = 5$. Although, for the Adult database, this value is not as alarming and is close to 60%.

3.4.3 Drop in Anonymity

Our next goal is to measure the drop in anonymity due to the independent release attack. To achieve this, we first take a closer look at the way these schemes work. As described in the earlier sections, the basic paradigm in partitioning-based anonymization schemes is to partition the data such that each partition size is at least k. The methodology behind partitioning and then

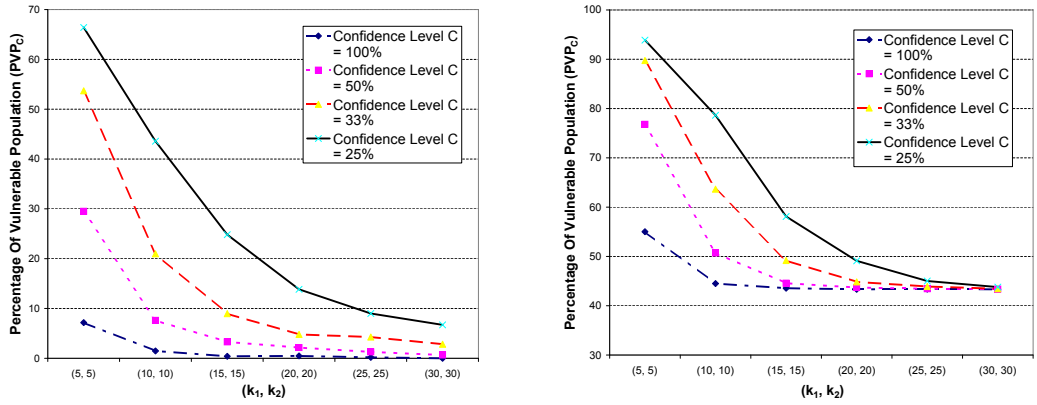


Fig. 3.2: Severity of the Independent Release Attack - Partial Breach (c) Adult Database and (d) IPUMS Database.

summarizing varies from scheme to scheme. The minimum partition-size (k) is thus used as a measure of anonymity offered by these solutions. However, the effective (or true) anonymity supported by these solutions is far less than the presumed anonymity k (refer to the discussion in Section 3.3.1).

Figure 3.3 plots the average partition sizes and the average effective anonymities for the overlapping population. Here again, we only consider the scenario where both the overlapping subsets are anonymized using Mondrian multidimensional technique. Observe that the effective anonymity is much less than the partition size for both the data subsets. Also, note that these techniques result in partition sizes that are much larger than the minimum required of k. For example, the average partition size observed in the IPUMS database for $k = 5$ is close to 40. To satisfy the k-anonymity definition, there is no need for any partition to be larger than $2k + 1$. The reasoning for this is straightforward as splitting a partition of size greater than $2k + 1$ into two results in partitions of size at least k. Additionally, splitting any partition of size $2k + 1$ or more only results in preserving more information. The culprit behind the large average partition

sizes is generalization-based on user-defined hierarchies. Since generalization-based partitioning cannot be controlled at finer levels, the resulting partition sizes tend to be much larger than the minimum required value.

For each individual in the overlapping population, the effective prior anonymity is equal to the effective anonymity. We define the average effective prior anonymity with respect to a release as effective prior anonymities averaged over the individuals in the overlapping population. Similarly, the average effective posterior anonymity is the effective posterior anonymities averaged over the individuals in the overlapping population. The difference between the average effective prior anonymity and the average effective posterior anonymity gives the average drop in effective anonymity occurring due to the independent release attack. Figure 3.4 plots the average effective prior anonymities and the average effective posterior anonymities for the overlapping population. Observe that the average effective posterior anonymity is much less than the average effective prior anonymity for both subsets. Also note that we measure drop in anonymities by using effective anonymities instead of presumed anonymities. The situation only gets worse (drops get larger) when presumed anonymities are used.

3.4.4 ℓ-diversity and t-closeness

We now consider the ℓ-diversity and t-closeness extensions to the original k-anonymity definition. The goal again is to quantify the severity of the independent release attack by measuring the extent to which a partial breach occurs with varying levels of adversary confidence levels. Figure 3.5 plots the percentage of vulnerable population for whom the independent release attack leads to a partial breach for the IPUMS databases. Here again, we anonymize both subsets

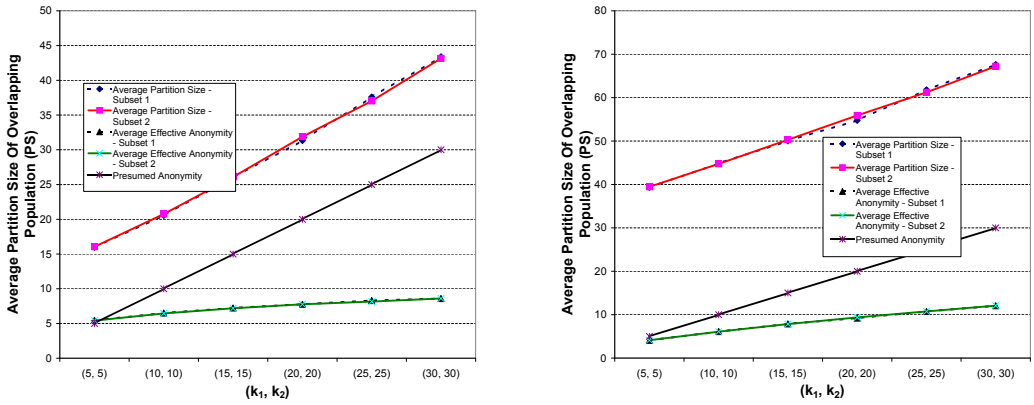


Fig. 3.3: Comparison of Presumed Anonymity, Actual Partition Sizes, and Effective Anonymity **(**a) Adult Database and **(**b) IPUMS Database.

of the database with the same definition of privacy. We use the mondrian multidimensional k-anonymity with additional constraints as defined by ℓ-diversity and t-closeness. Figure 3.5(a) plots the result for the ℓ-diversity using the same ℓ value for both the subsets ($\ell_1 = \ell_2$) and with $k = 10$. Figure 3.5(b) plots the same for t-closeness. Even though these extended definitions seem to perform better than the original k-anonymity definition, they still lead to considerable breach in case of an independent release attack. This result is fairly intuitive in the case of ℓ-diversity. Consider the definition of ℓ-diversity: the sensitive value set corresponding to each partition should be "well" (ℓ) diverse. However, there is no guarantee that the intersection of two well diverse sets leads to a well diverse set. t-closeness fares similarly. Also, both these definitions tend to force larger partition sizes, thus resulting in heavy information loss. Figure 3.5(c) plots the average partition sizes of the individuals corresponding to the overlapping population. It compares the partition sizes observed for k-anonymity, ℓ-diversity, and t-closeness. For the IPUMS database, with a value of $k = 10$, k-anonymity produces partitions with an average

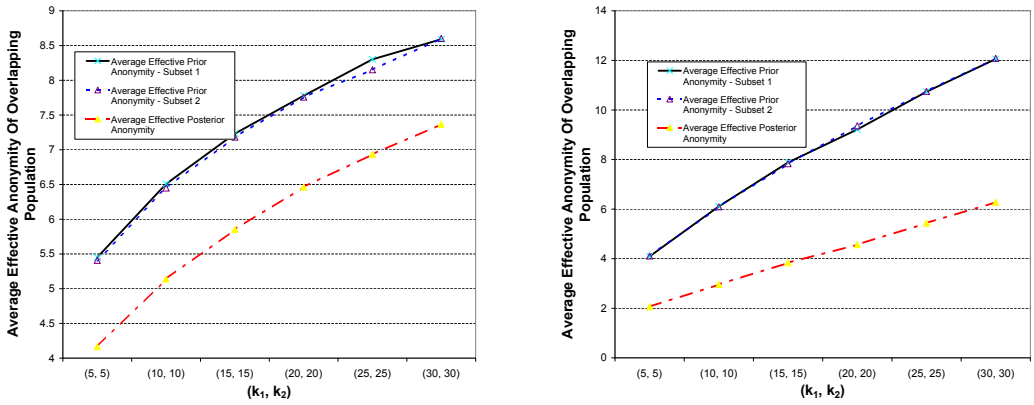


Fig. 3.4: Average Drop in Effective Anonymity **(**a) Adult Database and **(**b) IPUMS Database.

partition size of 45. While, for the same value of $k = 10$, with a value of $l = 5$, the average partition size obtained was close to 450. The partition sizes for t-closeness get even worse, where a combination of $k = 10$ and $t = 0.4$ yield partitions of average size close to 1300.

3.4.5 Role of Sensitive Attribute Domain

In all of the above experiments we use the "Occupation" (occupation code of the individual) as the sensitive attribute for both Adult and IPUMS databases as shown in Tables 3.5 and 3.6. The domain size of the Occupation attribute in the Adult database is 14, whereas, the domain size in the IPUMS database is 247. One of the plausible reasons for the attack to be more severe in case of the IPUMS database is the size of the sensitive attribute domain. This is because most of partition sizes are way larger than the minimum value required, i.e. k. In case of the Adult database, it is possible that the sensitive value set corresponding to every partition contains all the possible values in the domain. This implies that an intersection of two sensitive

value sets results in a set of size close to the size of the domain. Thus, it is possible that independent release attack will be less effective in cases where the sensitive attribute domain size is less than the average partition size. As a result, one can conclude that in cases where the sensitive attribute domain size is large (of the order of several hundreds) the independent release attack would be more severe. Also, most real-life databases have sensitive attributes with large domain sizes. For example, if we consider a typical hospital discharge database, an ICD9 code is used to describe the diagnosis given to the patient. The possible values for this code is a number from 1 to 999 ICD9 (2008) indicating the code for the specific patient diagnosis. In other cases, the sensitive attribute domain sizes tend be larger than this. The conjecture is that as the number of possible sensitive values increases, the intersection of two different sets results in a less diverse set.

In order to confirm this conjecture, we construct two new versions of the IPUMS database by replacing the sensitive attribute "Occupation" of each individual with "Industry" corresponding to the individual's work and "Income" corresponding to the total income of the individual. The domain sizes corresponding to these attributes are summarized in Table 3.7. The domain size for "Industry" attribute is 145, for the original "Occupation" attribute is 247 and that of "Income" is 471. Table 3.7 summarizes this. We ran the independent release attack on these new versions of the IPUMS database and compared it with the original. Figure 3.7 plots the average drop in effective anonymity for the overlapping population. Based on our conjecture, the drop in effective anonymity should increase with increase in the sensitive attribute domain size. Surprisingly we did not observe the trend we were expecting. The drop in effective anonymity in case of "Occupation" was less than when compared with "Industry". It turns out that the reason for this is the *actual* number of possible values for each sensitive attribute does not necessarily

be the same as the domain size, or in other words, the *total* number of possible values. So, a large sensitive attribute domain size does not guarantee that the number of possible values actually occurring is large. Instead, a simple entropy measure such as the Shannon's entropy could be used to measure the actual number of possible values. The entropy value for each of these attributes is listed in Table 3.7. Although the actual domain size for 'Occupation" attribute is larger, its entropy is less than that of the "Industry" attribute. Now, the conjecture is that as the entropy (or information content) of the sensitive attribute increases, the severity of independent release attack increases. Our result in Figure 3.7 confirms this. The average drop in effective anonymity increases with the entropy of the corresponding sensitive attribute domain since the non-sensitive attributes are the same for all the datasets.

Sensitive Attribute	**Domain Size**	**Entropy**
Occupation	247	4.30
Industry	145	4.35
Income	471	5.56

Table 3.7: Sensitive attribute changes in IPUMS Database

3.4.6 Number of Databases

In the above experiments we consider the scenario in which two anonymized releases contain information about overlapping population. As data publishing becomes more prevalent among organizations, it is possible that the number of anonymized releases available containing information about the same subset of people is more than just two. The adversary could use as many anonymized releases as possible to gather information about a target population and use the independent release attack to deduce the sensitive attribute values. In such a scenario, it is

interesting to see how the independent release attack performs. We first consider the percentage of vulnerable population with a confidence level of 100% ($PVP_{100\%}$). Figure 3.8(a) plots this for varying number ($n = 2, 3, 4$) of anonymized releases available to adversary. Here again, we build n overlapping subsets of the IPUMS database by fixing the overlapping population at 5000. It can be observed that the severity of the independent release attack increases with the increase in the number of anonymized releases available to the adversary. There is a significant increase in the percentage of vulnerable population with the increase in n, for small values of k. However, there seems to be no such significant increase for larger values of k. The reason for this is that the partition sizes for larger values of k tend to be large enough such that the presence of additional anonymized releases does not help the independent release attack anymore. Apart from the severity of the attack, it is interesting to see the effect of number of anonymized releases on the drop in effective anonymity. Figure 3.8(b) plots the average drop in effective anonymity for varying number ($n = 2, 3, 4$) of anonymized releases. Here again, we can observe that drop in effective anonymity increases with the increase in the number of anonymized releases. These results also indicate that if the anonymized releases correspond to fairly larger values of k, there is only a limited gain in the information gathered by the adversary by collecting additional releases.

3.5 Conclusions

Our experimental study proves that several currently proposed partitioning-based anonymization schemes, including k-anonymity and its variants, are vulnerable to independent release attacks. For two different implementations of k-anonymity, we found that sensitive information of a significant percentage of population could be compromised. These methods seem to mitigate

independent release attacks to some extent by producing artificially large clusters. Refining the algorithms to produce finer clusters would not help as it will only increase the severity of the independent release attack. The extended definitions of ℓ-diversity and t-closeness fare better than the original k-anonymity definition but still lead to considerable breach. Additionally, these schemes lead to huge partition sizes and thus result in heavy information loss. Our results indicate that the severity of the attack increases with the increase in entropy of sensitive attribute domain. Further, the severity of the attack increases with the available number of independent releases.

This work has been published in the 2008 ACM International Conference on Knowledge Discovery and Data Mining Conference (SIGKDD) Ganta et al. (2008a). A more comprehensive report of the study is made available through a techreport Ganta et al. (2008b).

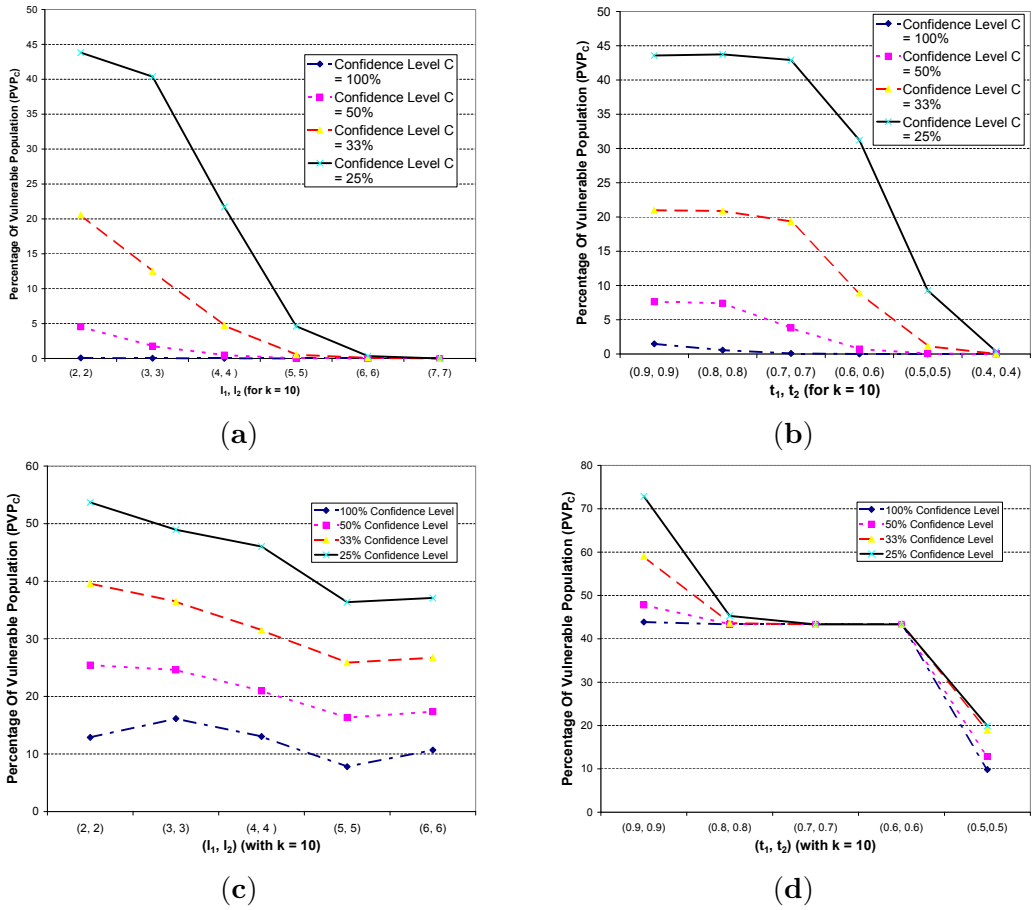


Fig. 3.5: Severity of Independent Release Attack - **(**a)(c)ℓ-diversity and **(**b)(d)t-closeness **(**a)(b) Adult Database and **(**c)(d) IPUMS Database

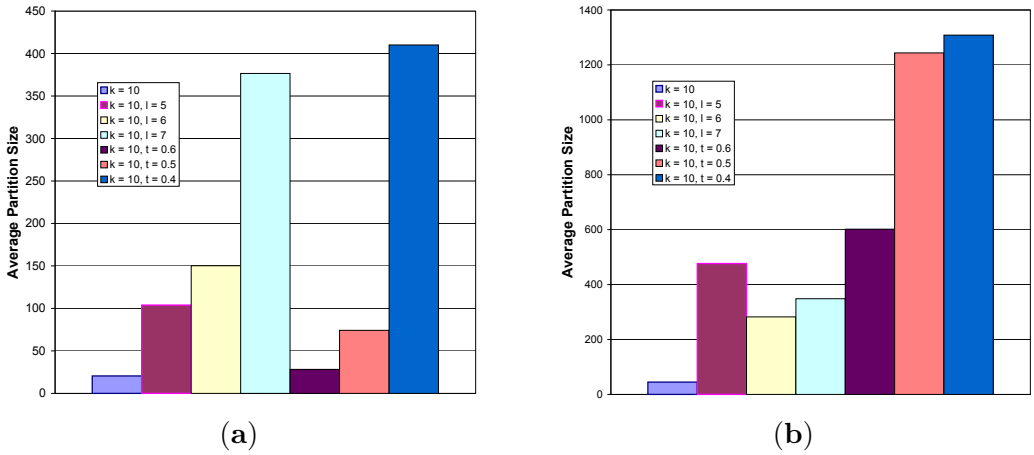


Fig. 3.6: Average Partition Sizes for ℓ-diversity and t-closeness **(**a) Adult Database and **(**b) IPUMS Database

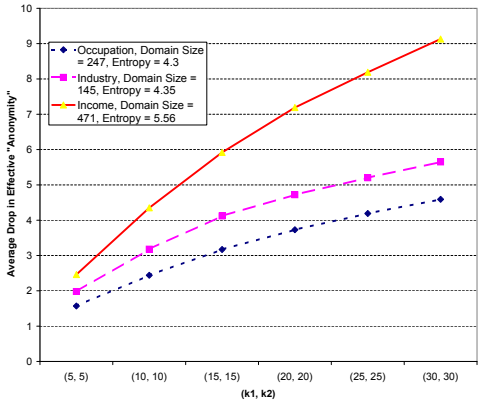


Fig. 3.7: Effect of Sensitive attribute Domain - IPUMS Database.

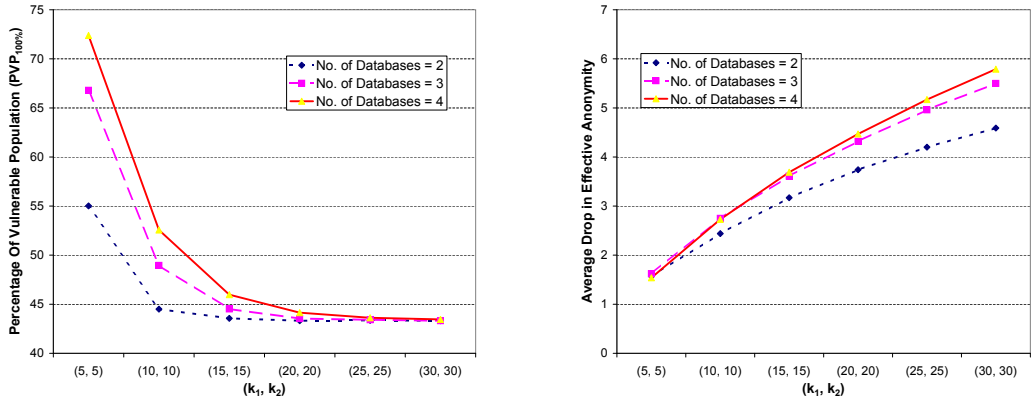


Fig. 3.8: Effect of Number of Independent Releases - IPUMS Database **(**a) Percentage of Vulnerable Population and **(**b) Drop in Effective Anonymity

Chapter 4

Information Fusion Attack 2: Web Based Attack

In this chapter, we investigate the problem of Web based Attack. Section 4.1 introduces the problem. Section 4.2 discusses related work and Section 4.3 elaborates on the attack model. In Section 4.4, we formulate the problem of Fusion Resilient Enterprise Data Anonymization. We then present our solution strategy in Section 4.5. Section 4.6 presents experimental results by demonstrating the attack on a real data set. Section 4.7 provides the conclusions.

4.1 Problem

Recall that partitioning-based anonymization schemes classify data attributes into identifier, non-sensitive and sensitive attributes. Based on this classification, existing solutions assume that the *identifier attributes* are stripped *prior* to the anonymization process. This is done under the assumption that identifier attributes are either not present in the database or are not necessary for the intended purpose of the release. This assumption however is too restrictive and does not consider specific data publishing scenarios such as internal releases Ganta & Acharya (2008a). Consider a sample patient database maintained by a healthcare insurance enterprise as depicted in Table 4.1. The data contains names of all the customers along with certain *non-sensitive* and *sensitive* information. The non-sensitive attributes are: *Previous Visit Index (Visit Index)* to indicate the the number of care visits made by the customer in the past, *Expenditure Index (Exp Index)* to indicate the amount spent on the customer, *Customer Valuation (Valuation)* to indicate

the valuation of the customer. The only *sensitive* attribute, *Customer Personal Income (Income)*, corresponds to estimated customer's personal income. Databases such as this are used to evaluate customers and determine the policy premium etc. The internal *release* of such data along with the explicit identifiers (*Customer Names*) is a necessity for several enterprise operations such as accounting, record keeping etc. However, such a release should not disclose sensitive information, in this case, *Customer Personal Income*. Note that trivial solutions such as use of pseudonyms are not viable in such scenarios. The key properties here are:

- The inclusion of identifying information is necessary for the *release* to serve the intended purpose.
- The focus is on protecting against attribute disclosure (as opposed to identity disclosure) even in the presence of explicit identifiers.

Name	Visit Index	Exp Index	Valuation	Income
Alice	8	7	4	91,250
Bob	5	4	4	74, 340
Christine	4	5	5	75,123
Robert	9	8	9	98,230

Table 4.1: Typical Customer Data in Medical Insurance Organizations

Now, consider that the publisher uses partitioning-based anonymization scheme such as k-anonymity to anonymize the data and releases it. Table 4.2 shows the k-anonymized version of this data.

This version is deemed *safe* and is released internally within the organization. Now, consider the scenario in which an adversary employee *Bob* is granted access to this anonymized release. Note that the release does not give *Bob* access to the sensitive information i.e customer

Name	Visit Index	Exp Index	Valuation	Income
Alice	[5-10]	[5-10]	[1-5]	-
Bob	[5-10]	[1-5]	[1-5]	-
Christine	[1-5]	[1-5]	[1-5]	-
Robert	[5-10]	[5-10]	[5-10]	-

Table 4.2: Anonymized Customer Data

Name	Employment	Property Holdings
Alice	CEO, Deutsche Bank	3560
Bob	Manager, Verizon	1200
Christine	Assistant, NYU	720
Robert	CEO, Microsoft	5430

Table 4.3: Potential Auxiliary Data

personal income data. However, he has access to non-sensitive information (customer valuation, investment volume etc) along with the customer identifiers. *Bob*'s goal is to use this information to estimate the customer personal income values. To achieve this, he uses the customer names present in the release to search for auxiliary information available on the web which will help him estimate their personal income. For example, he could collect information about the customer's *Employment*, *Property Holdings* etc. Example of such data collected from the web is shown in Table 4.3. Now, by putting together this information with the anonymized release the adversary can estimate the sensitive customer personal income information. For this, the adversary can use a range of simple knowledge based methods to sophisticated information fusion techniques such as *fuzzy inferencing* Kosko (2005). For example, let's say the income range for all the customers is [$40000 - $100000] and could be divided into three classes *Low* [$40000 - $60000], *Medium* [$60000 - $80000], and *High* [$80000 - $100000]. Now, consider the customer *Robert*. With an estimated valuation falling in the highest range [5-10], *Bob* concludes that *Robert* falls into the highest income category [$80000 - $100000]. By looking at his employment and property

holdings (and possibly other auxiliary information), *Bob* can further improve his estimate and conclude that *Robert* falls into upper category [$90000 - $100000] of the *High* income class. Based on this, he estimates that *Robert's* salary is the average of range [$90000 - $100000] i.e $95000.

This example demonstrates, how an adversary could obtain a close estimate of *Robert's* actual income by using the auxiliary information obtained from the web. This attack involves inferential attribute disclosure based on available identifiers and possibly involves a *human-in-the-loop*. We assume that the intruder is an *insider* who is given or otherwise acquires access to the anonymized data. Thus, the intruder has access to individual identifiers that can be used to index into the web and other data sources. The intruder is also assumed to have domain knowledge about the data to perform information fusion.

4.2 Related Work

Inferential disclosure has been extensively studied in the statistical literature G.T.Duncan & Lambert (1986) G.T.Duncan (1990). In the computer science community, Aggarwal et al. (2006) provide a first formal treatment of inferential attribute disclosure. However, the attack model does not involve any auxiliary information and uses association rule mining to infer sensitive data from the anonymized release. Martin et al. (2007) provides a formal treatment of adversarial background knowledge. They propose a language for expressing the adversary's knowledge based on conjunctive propositions. More recently, Chen et al. (2007) have attempted to fill this gap, by proposing an extension to the same language based framework. However, these models do not consider auxiliary information obtained *using identifying information present in the anonymized release*. Orthogonal to these works, Wong et al. (2007) prove that adversary's

knowledge of the anonymization algorithm could lead to a privacy breach. Our work is critically different from these studies as we consider inferential attribute disclosure based on auxiliary information obtained from the web using identifier information available in the data.

4.3 Fuzzy Inferencing

We use *fuzzy inferencing* to build an information fusion system to put together the anonymized release with web-based auxiliary information. *Fuzzy Inference* is a well-studied paradigm based on *fuzzy logic*, *fuzzy if-then rules* and *fuzzy reasoning*. It provides a mechanism to *map* a set of *inputs* to a set of *outputs* using a set of *rules*. We refer the reader to Kosko (2005) for an introduction to fuzzy inference systems. The first step involved in creating a fuzzy inference system is to determine the *inputs* and *outputs*. In our web-based information-fusion attack, the inputs include all the data attributes available to the adversary through: 1. The anonymized release and 2. The auxiliary data collected through the web. In our running example from Section 4.1, the attributes *Visit Index, Exp Index, Customer Valuation* from the anonymized release in Table 4.2 form the first half of inputs to the information fusion system. The attributes *Employment, Property Holdings* collected from the web form the second half of inputs. The output consists of single attribute, *Customer Personal Income*, which the adversary intends to estimate. In the second step, the adversary defines *fuzzy-set definitions* for each of the input and output attributes. He then uses domain knowledge to formulate a set of *rules* mapping the input fuzzy sets to the output fuzzy sets. Figure 4.1 illustrates the system.

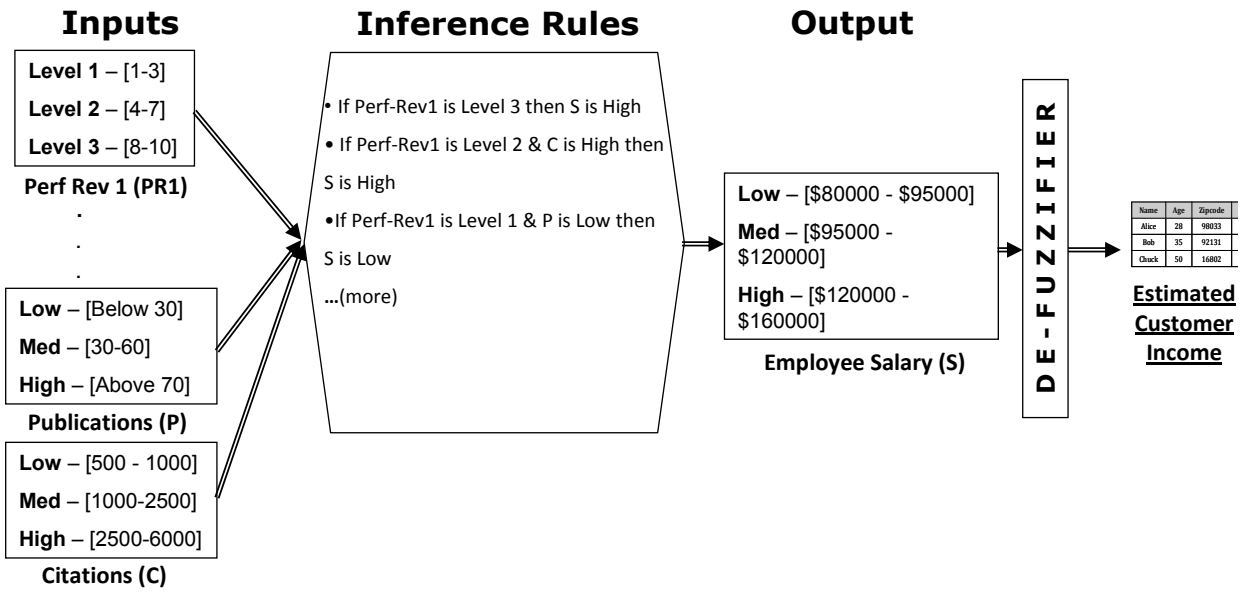


Fig. 4.1: Fuzzy Inference System

4.4 Problem Formulation

We now formulate the problem of *Fusion Resilient Enterprise Data Anonymization* to address web-based attacks. Since it is not possible to quantify the amount of auxiliary information the adversary can collect, it is not practical to completely prevent such attacks. However, by estimating the auxiliary information that an adversary could collect, we can *minimize the extent of privacy breach* in case of such an attack. This forms the primary goal of our problem formulation: For a given sensitive dataset, we need to find an anonymization such that the release causes minimum breach in case of a fusion attack. On the other hand, one of the important factors involved in data anonymization is the *utility* of the release Bayardo & Aggarwal (2005) LeFevre et al. (2006a). The utility of an anonymized release is a measure of usefulness of the release for the intended purpose such as a specific task to be performed on the data Ex. Classification etc. Several standard measures such as Bayardo & Aggarwal (2005) have been proposed in the

literature to compute data utility. Hence, the secondary goal of our problem formulation is to maximize data utility. With these goals in hand, we proceed to formulate the overall goal as follows:

Let $P = \{p_{ij}\}_{m \times n}$ be a sensitive private dataset defined over a finite set of attributes $\{P_1, P_2, \ldots, P_n\}$.

Let $Q = \{q_{ij}\}_{r \times s}$ be the auxiliary data gathered by the *intruder* from the web over a set of attributes $\{Q_1, Q_2, \ldots, Q_s\}$.

Now, let P' be a *candidate* anonymization of P.

Let F be an *information fusion* system that takes in P' and Q as inputs and produces $\hat{P}$, an estimate of P.

Let U be a measure of utility of P'.

Goal : The goal of *Fusion Resilient Enterprise Data Anonymization* is to compute *a* P' from P such that:

1. P' is *resilient* to *Web-based Information Fusion Attacks*.

2. The utility U offered by P' meets the release requirements.

To formulate the problem based on the above goal, we need to quantify the *resilience* to web-based information-fusion attacks. We define this using the following definitions:

Definition 10 (Dissimilarity ($D_1 \circ D_2$)) For two datasets D_1 and D_2 representing the same set of individuals and the same set of attributes, $D_1 \circ D_2$ is a measure of *dissimilarity* between them.

For two datasets $\{D_1\}_{m\times n}$ and $\{D_2\}_{m\times n}$ representing the same set of individuals, we compute the dissimilarity using mean square distance D_1 and D_2:

$$D_1 \circ D_2 = \frac{1}{m} * Tr((D_1 - D_2)^T (D_1 - D_2))$$

where m is the total number of records in each database and $Tr(A)$ of a matrix A is the trace of A, i.e the sum of the elements of the main diagonal.

As defined earlier, $\hat{P}$ is an estimate of P made by the adversary based on a candidate release P' and web-based auxiliary data Q using the information fusion system F.

$$\hat{P} = F(P', Q)$$

In order for privacy of P to be protected, the dissimilarity between P and the estimate made by the adversary, $\hat{P}$, needs to be *large*. The more the dissimilarity $P \circ \hat{P}$, the better protected P is. Also, the dissimilarity between P and $\hat{P}$ *quantifies* the *protection* offered by the corresponding P' against information fusion attacks. Based on this, we now define a *Fusion Resilient Anonymization* as:

Definition 11 (Fusion Resilient Anonymization) An anonymization P' of a given sensitive data P is resilient to fusion attacks if the dissimilarity $(P \circ \hat{P})$ between $\hat{P}$ and P is above a certain threshold value T_p.

So, for a candidate anonymization P' to be a *safe* release, the corresponding $(P \circ \hat{P})$ needs to be above a certain threshold value T_p. It is obvious to note that, among all the possible anonymizations (P's) that satisfy this property, the one that has maximum value of $(P \circ \hat{P})$ offers

maximum protection. So, for the anonymization P' to offer maximum resilience to web-based information fusion attacks, the *dissimilarity* $(P \circ \hat{P})$ needs to be maximized.

Recall that in addition to maximizing the protection against information-fusion attacks, the utility of the release (U), should be maximized. Let W_1 and W_2 be the weights assigned by the publisher for privacy protection against information fusion attacks and data utility respectively. Now, the final objective can be stated as a *weighted sum of protection and utility* of the form:

$$W_1 * (P \circ \hat{P}) + W_2 * U$$

Now, the problem can be stated as,

Problem : Given a private dataset P, web-based data Q and an information-fusion system F, find the fusion resilient anonymization P' that maximizes $H = W_1 * (P \circ \hat{P}) + W_2 * U$, where $\hat{P}$ represents the estimate of P based on P' and Q using F.

In order to solve the above *optimization problem*, we need to find the *optimal* anonymization P' in the *solution space* containing all possible anonymizations $P's$ that satisfy the fusion-resilient-anonymization property defined earlier. One way to look at this *solution space* is to consider the set of all anonymizations possible by anonymizing P to different *levels*. Note that the definition of *Anonymization Level* depends on the specific anonymization scheme to be employed. For example, in k-anonymization, the value of k represents the anonymization level. The more the value of k is, the more the anonymization level. As mentioned in Section 1, in our work, we use k-anonymization as the basic anonymization scheme. For a given dataset P, let i denote the anonymization level and P'_i denote the release obtained by anonymizing P to level i. We use the *discernibility metric* defined in Bayardo & Aggarwal (2005) to measure the utility of

a k-anonymized data set. The metric can be mathematically stated as follows.

$$C_{DM}(g,k) = \sum_{\forall |E| \geq k} |E|^2 + \sum_{\forall |E| < k} |D| * |E|$$

where E refers to the *clusters* or *equivalence classes* of the data set induced by k-anonymization of g using the value k. The reader is referred to the original paper for further details. Based on the above definition, let the utility of P'_i be denoted by U_i. The optimization function H can now be defined based on anonymization level i as:

$$H_i = W_1 * (P \circ \hat{P}_i) + W_2 * U_i$$

Let T_u be the minimum utility required for the release. Now, the above generic problem statement can be instantiated as:

Problem Statement: Find $P'_{i_{opt}}$, such that

$$H_{i_{opt}} = \max_{\forall i} H_i$$

where, $(P \circ \hat{P}_i) \geq T_p$ and $U_i \geq T_u$.

4.5 Solution

We now propose a simple iterative algorithm to find the *fusion resilient anonymization* for a given sensitive dataset. The strategy is to take any basic anonymization scheme such as k-anonymization and *incrementally* anonymize the data. The level of anonymization is increased in steps (increase k in steps), until the utility of the release falls below a threshold. In each step,

the web-based fusion attack is simulated to find whether the resulting candidate anonymization offers enough protection. If yes, the candidate anonymization is retained, otherwise it is discarded. This results in a set of all *candidate* anonymizations present in the *solution space*. We then search for the optimal anonymization level that offers the maximum weighted sum of protection and utility. Figure 4.2 illustrates our approach.

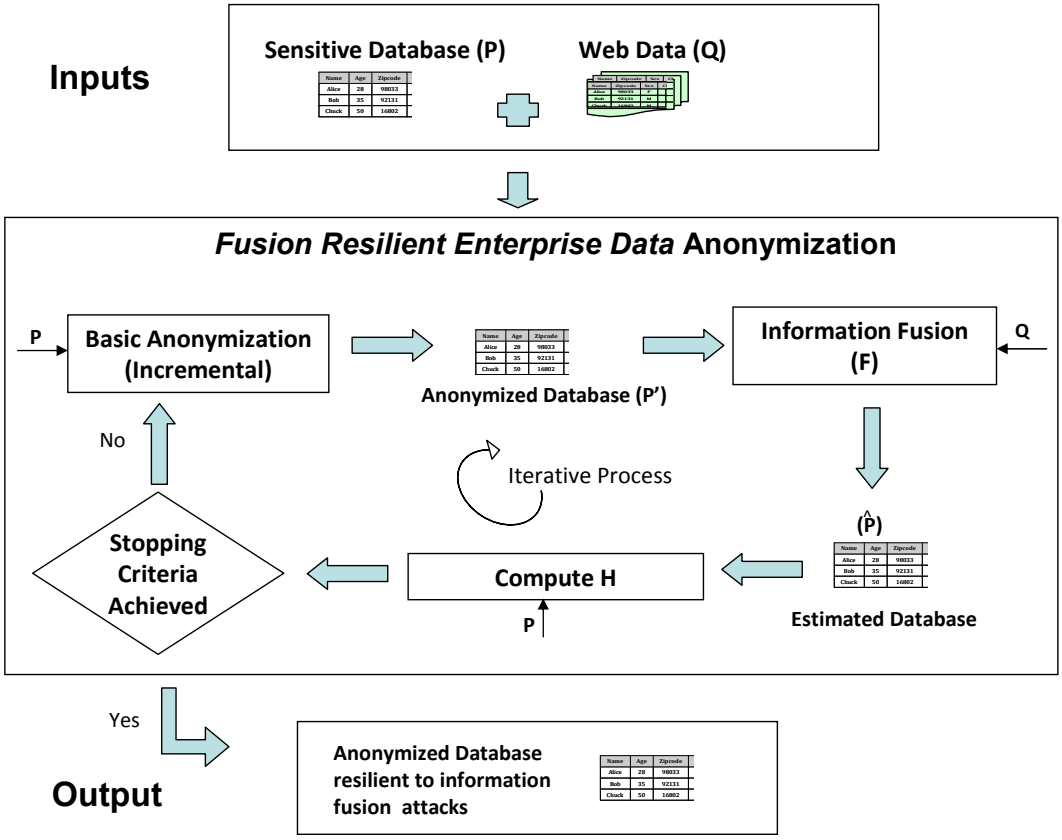


Fig. 4.2: Fusion Resilient Enterprise Data Anonymization

Algorithm 2 presents this solution in procedural format as FRED_Anonymization (Fusion Resilient Enterprise Data Anonymization). The algorithm uses the Basic_Anonymization procedure that takes a sensitive data and level of anonymization as inputs and produces an anonymization of the input data to the corresponding anonymization level. For this, any basic anonymization algorithm such as the ones proposed in Domingo-Ferrer & Mateo-Sanz (2002) LeFevre et al. (2006a) can be used to generate a k-anonymization. Note that in case of k-anonymization the minimal level of anonymization is achieved by using the value $k = 2$. The algorithm uses the Basic_Anonymization procedure to anonymize the sensitive data for increasing values of the anonymization level $(level)$. The stopping condition for this loop is achieved when the utility of anonymized result (P') denoted by U_{level} falls below the threshold T_u. In each iteration, the algorithm simulates an information fusion attack to produce the estimate an adversary could obtain ($\hat{P}_{level}$). The *dissimilarity* between the estimated values $\hat{P}_{level}$ and the original values P is computed using the procedure Dissimilarity_Measure which takes two datasets as input and outputs the dissimilarity value as described in Section 4. At this point, the dissimilarity is compared against a threshold value T_p to check if the anonymization offers enough protection against information fusion attacks. If yes, the weighted sum of dissimilarity and utility is computed and stored as $H(i)$. Finally, the algorithm searches for the anonymization level i that offers the maximum value for the weighted sum of protection and utility H_{max}. The anonymization $P'_{i_{opt}}$ corresponding to H_{max} is the fusion resilient anonymization of the original data that offers maximum weighted protection as well as utility.

Algorithm 2 FRED_Anonymization

$P \leftarrow$ Sensitive Data
$Q \leftarrow$ Web Data
$F \leftarrow$ Information Fusion System
$T_p \leftarrow$ Protection Threshold
$T_u \leftarrow$ Utility Threshold
$W_1 \leftarrow$ Protection Weight
$W_2 \leftarrow$ Utility Weight
$level \leftarrow -1$
$i \leftarrow 0$
repeat
  $level \leftarrow level + 1$
  $P'_{level} \leftarrow$ Basic_Anonymization(P, level)
  $\hat{P}_{level} \leftarrow F(P'_{level}, Q)$
  $P \circ \hat{P}_{level} \leftarrow$ Dissimilarity_Measure$(P, \hat{P}_{level})$
  $U_{level} \leftarrow$ Utility(P'_{level})
  if $(P \circ \hat{P}_i) \geq T_p$ **then**
    $H(i) \leftarrow W_1 * (P \circ \hat{P}_{level}) + W_2 * U_{level}$
    $i \leftarrow i + 1$
  end if
until $U_{level} \geq T_u$
$i_{max} \leftarrow i - 1$
$H_{max} \leftarrow H(0)$
for $i = 1$ to $i = i_{max}$ **do**
  if $H(i) \geq H_{max}$ **then**
    $i_{opt} \leftarrow i$
  end if
end for
return $P'_{i_{opt}}$

4.6 Experimental Results

In this section we present experimental results by demonstrating the *web-based information-fusion attack* on a real-life dataset. The goals here are to quantify the information gained by the adversary through information fusion and demonstrate the FRED_Anonymization algorithm.

4.6.1 Setup

The sensitive data (P) is collected from a real-life enterprise (a public university) and contains salary information and performance review numbers of the employees (faculty). The employee *Salary* is the *sensitive* attribute while the performance review numbers are the *non-sensitive attributes*. The data is anonymized (P') so as to *suppress* all of the salary information and k-anonymize the non-sensitive attributes using *microaggregation* based k-anonymization proposed in Domingo-Ferrer & Mateo-Sanz (2002). The external data(Q) is collected from the employee web pages and external links from there. Based on domain knowledge, we formulate a simplistic set of knowledge rules to fuse P' and Q and build a fuzzy inference system to estimate the employee salary as illustrated in Figure 4.1. All the rules are assigned uniform weights.

All the experiments were implemented using Matlab on a PC with Intel Pentium 4 (1.8GHz) processor and 1GB of RAM running Microsoft Windows XP.

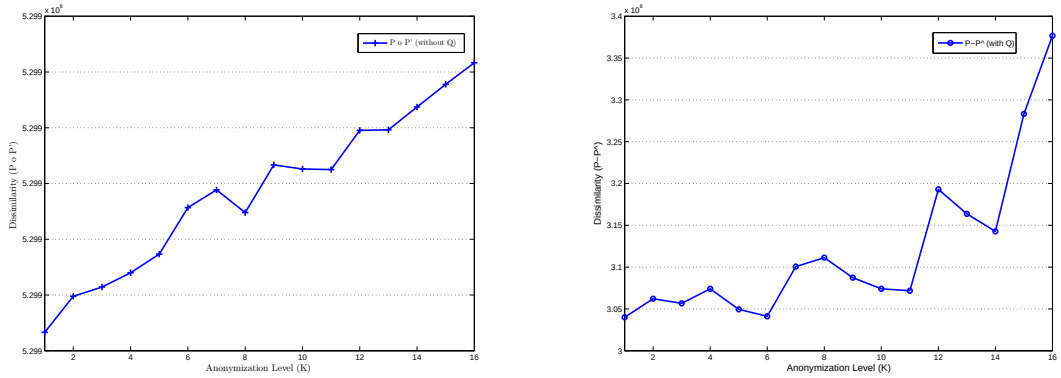
4.6.2 Information Gain

Our first study aims to quantify the *information gain* obtained by the attacker in estimating the sensitive data P, by introducing web-based auxiliary information Q. Consider the adversary's knowledge of the original data at two stages 1. *Before* information fusion, and 2. *After* information fusion. Recall that to start with, the adversary has access to the anonymized release

P'. The adversary then collects Q and fuses this with P' to obtain $\hat{P}$. So, before performing information fusion, the adversary's (best) knowledge about the original data is the anonymized version itself, i.e P' (in the absence of Q). In this case, we have the dissimilarity between the original and the adversary's estimate $(P \circ \hat{P}) == (P \circ P')$. Figure 4.3(a) plots this $(P \circ P')$ for increasing values of k. It is not surprising to observe that the *dissimilarity* increases as k increases, since the *level* of anonymization increases with k. After performing information fusion, the adversary obtains $\hat{P}$ by fusing P' with Q using F. Figure 4.3(b) plots this $(P \circ \hat{P})$ for increasing values of k. Notice that $(P \circ \hat{P})$ is lesser than $(P \circ P')$ for all values of k. In other words, the estimate made by the attacker $(\hat{P})$ *after* information fusion is closer to (P) than when compared to the estimate available *before* information fusion (P'). The difference between $(P \circ P')$ and $(P \circ \hat{P})$ is precisely the amount of information gained by the adversary through information fusion. Hence, the *Information Gain* G of the adversary is the difference between the closeness of the estimates available before and after information fusion.

$$G = (P \circ P') - (P \circ \hat{P})$$

Figure 4.4 plots G for increasing values of k. It is interesting to observe that G does not necessarily increase with k. This implies that as the level of anonymization increases, the information gained by the attacker decreases. The reason for this is that as the level of anonymization increases, the input (P') to the information fusion system gets worse and thus forces the system to output incrementally bad estimates.



(**a**) (**b**)

Fig. 4.3: (a) Before Information Fusion $(P \circ P')$ (b) After Information Fusion $(P \circ \hat{P})$

4.6.3 Optimal Anonymization

We now study the fusion resilient enterprise data anonymization that leads to maximum weighted sum of protection and utility as formulated in Section 4. We use the *discernibility metric* defined in Bayardo & Aggarwal (2005) to measure the utility of a k-anonymized data set. The basic idea here is to assign each data sample (or vector) a *cost* based on the number of data vectors it is indistinguishable from, or in other words, the size of the cluster it falls into. If the cluster size it falls into is greater than k, then the cost assigned is equal to the size of the cluster. If the cluster size is less than k, then the cost is much severe (since it does not adhere to the definition of k-anonymity) and is equal to the product of the size of the whole data set and the size of the cluster.

$$C_i = \left\{ \begin{array}{ll} |E|^2 & if|E| \geq k \\ |D| * |E| & otherwise \end{array} \right\}$$

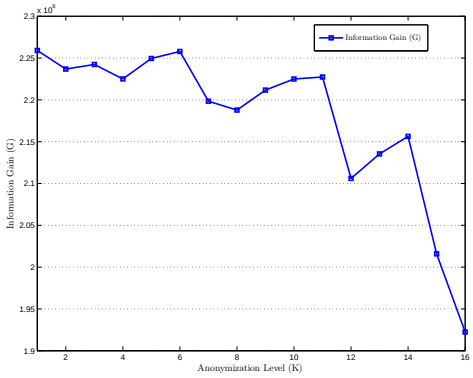


Fig. 4.4: Information Gain (G)

Using this definition, we define the utility of the data set $U = \{u_{i1}\}_{m\times 1}$ as a column matrix where each entry is the inverse of the cost assigned to the corresponding data point.

$$u_{i1} = 1/C_i$$

To show how utility of the release varies with increasing level of anonymization (increasing values of k), we calculate the utility of the entire release using the discernibility definition Bayardo & Aggarwal (2005) as:

$$C_{DM}(k) = \sum_{\forall |E| \geq k} |E|^2 + \sum_{\forall |E| < k} |D| * |E|$$

$$U_k = 1/C_{DM}(k)$$

Figure 4.5 plots U_k for increasing values of k. It is straight-forward to observe that utility of data decreases as k increases. The goal now is to find the optimal k value such that the resulting

anonymization offers maximum weighted sum of privacy protection and utility formulated as:

$$H = \frac{1}{m} * Tr((P \circ \hat{P})^T W_1 (P \circ \hat{P})) + \frac{1}{m} * Tr(U^T W_2 U)$$

We establish the threshold values for protection and utility as $T_p = 3.075$ $T_u = 0.0018$ based on experimental observations. For these threshold values, we obtain the solution space of $k = 7$ to 14. We assign equal weights to privacy protection and utility i.e $W_1 = W_2 = 0.5$, $W_i = 0.5_{m \times n}$i.e . Based on this setup, Figure 4.6 plots H for increasing values of k within the solution space. By running an optimization for the maximum value of H, we obtain the result $k = 12$. This is the optimal k value that provides the maximum weighted sum of protection and utility.

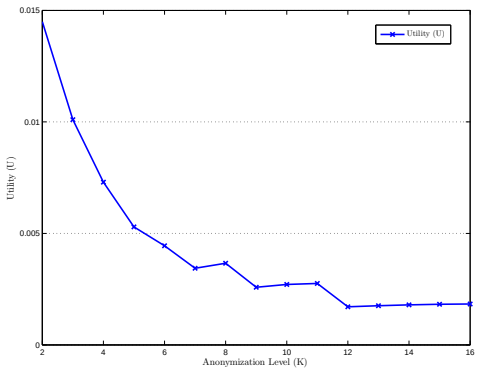


Fig. 4.5: Utility U_k

4.7 Conclusions

Our experimental study proves that partitioning-based anonymization schemes such as k-anonymity are vulnerable to web based information fusion attacks. We found that sensitive

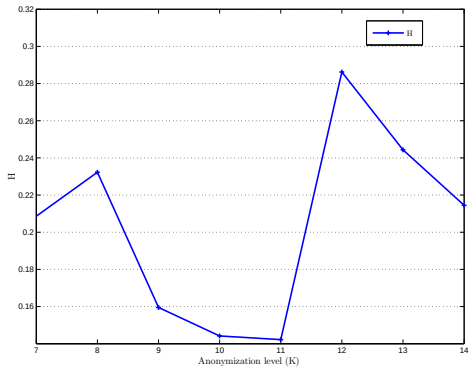


Fig. 4.6: Weighted Sum Of Protection And Utility H_k

information can be inferred with significant levels of precision in case of such attacks. While it is not possible to entirely prevent fusion based privacy attacks, one can minimize the extent of breach possible through intelligent data anonymization. Our problem formulation achieves this goal by incremental data anonymization until a threshold value of privacy is attained. An avenue for future work would be study sophisticated information fusion methodologies to detect inferential attribute disclosure. It would be interesting to look at specific sources on the web that could help an adversary in finding personal information and carrying out such attacks.

This work has been published in the 2008 ACM Symposium on Applied Computing (SAC) Ganta & Acharya (2008a) and 2008 IEEE International Conference on Data Engineering (ICDE) Ganta & Acharya (2008b). A more comprehensive report of the study is made available through a techreport Ganta & Acharya (2008c).

Chapter 5

Privacy Preserving Sharing

In the last few chapters we focused on privacy preserving data publishing where a central organization releases anonymized version of sensitive data for public use. Now, we move our focus to the problem of privacy preserving data sharing where an organization aims to share sensitive data with other authorized organizations without any modifications. Consider for example, a group of collaborating hospitals that would like to share their corresponding patient records for providing better service and promote interoperability. The organizations trust each other and would like to have access to sensitive data as-it-is without any modifications. This setting is depicted in Figure 5.1. The key difference here is that the goal in data publishing is to release global statistics of the data whereas in data sharing the goal is unmodified access to sensitive data among trusted parties. The challenge, however, is to ensure that privacy policies associated with the shared data are honored by the sharing mechanism. For example, say a source organization has a privacy policy associated with its patient records. When the source shares this data with an authorized party, it needs a guarantee that all the applicable privacy rules according to the source's policies are applied at the recipient as well. So, the goal in privacy preserving data sharing is to devise a mechanism that ensures distributed compliance of privacy policies.

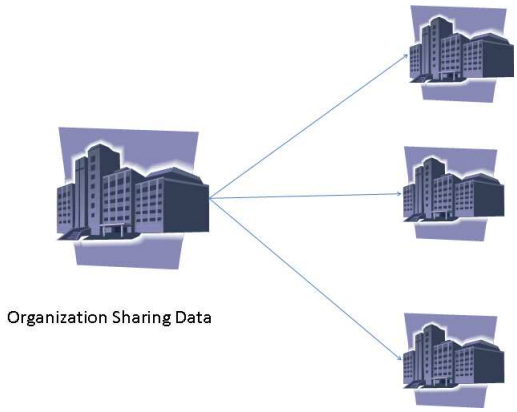


Fig. 5.1: Data Sharing

5.1 Related Work

The problem of privacy preserving sharing of policy-regulated data has received little attention until recently. Agrawal et al. (2002) proposed the first design principles for building database systems that manage privacy policy-regulated data. The system is termed as *Hippocratic Database*. The design involves specification of privacy policies that describe the recipients - who gets access to what parts of data, purposes - the purpose for which access is granted and conditions - the conditions under which the access is granted. The policies are designed using a specification language that satisfies the P3P standard Cranor et al. (2002b), e.g. APPEL Cranor et al. (2002a). The design also provides other functionality such as support for maintaining audit trails Agrawal et al. (2004), query rewriting for disclosure limitation LeFevre et al. (2004), and data retention. Snodgrass et al. (2004) proposes schemes for auditing the operations of a database such that any tampering with the audit logs can be detected. Such a solution can guard against the databases manipulation of the audit logs, thus giving assurance of eventual post-breach detection.

5.2 Challenges

In this dissertation, we focus primarily on privacy preserving sharing of healthcare documents. Hence, we use the terms data and documents interchangeably in the rest of the text. Currently, healthcare organizations operate in isolation. Each organization maintains an independent clinical record database with disparate privacy policies governing access to these databases. However, several government studies have indicated the need for *interoperability* that allows seamless sharing of clinical records across organizational boundaries Policies in focus: Strengthening Healthcare (2006) Yasnoff et al. (2004) Re-engineering the Clinical Research Enterprise (2005) The Goals of Strategic Network (2005). Interoperability is expected to significantly improve the quality, expediency and efficiency of healthcare delivery while decreasing costs to patients and providers.

Towards this goal, the government initiated the formation of information centers called Regional Health Information Organizations (RHIOs) The Goals of Strategic Network (2005) The California Regional Health Information Organization (2006). A RHIO is an independent regional collaboration of healthcare entities that maintains a single privacy policy-regulated data management system. Figure 5.2 presents a typical RHIO environment. A limited number of RHIOs already exist today in the U.S. The California Regional Health Information Organization (2006). Each RHIO is expected to operate as an individual enterprise and have precisely defined privacy policies that govern access to its data.

The current challenge however is to achieve an interoperable healthcare system where a doctor located anywhere in the country can access a patient's medical documents from any RHIO in emergencies and authorized settings. But, several questions need to be answered to address

this problem. What if the privacy policies followed by the RHIOs sharing data are conflicting? Whose policies take precedence in such scenarios? Can a RHIO that is receiving data pass it on to a third RHIO? So, the goal is to achieve cross-enterprise data sharing that facilitates seamless sharing of sensitive data across enterprises that follow potentially conflicting privacy policies.

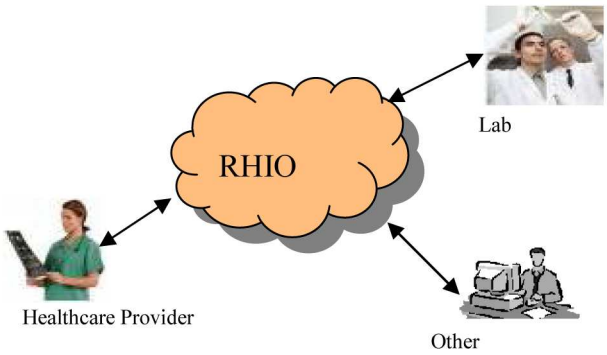


Fig. 5.2: Regional Health Information Organization(RHIO).

Chapter 6

Cross-Enterprise Document Sharing through Sticky Privacy Policies

In this chapter, we address the problem of cross-enterprise data sharing by designing a system for distributed enforcement of privacy policies. Section 6.1 introduces the problem and Section 6.2 presents the prior work done. Section 6.3 presents the design principles of the sticky privacy policies and Section 6.4 presents the system architecture. In Section 6.5, we provide a performance evaluation of our system. Section 6.6 provides the conclusions.

6.1 Problem

Let us consider an example to motivate the cross-enterprise data sharing problem. Consider a scenario where Sandra, a Californian patient at GoodSam Hospital in the CaliShield RHIO, goes in for treatment related to her nasal allergies. During her visit, she finds out that she has asthma and needs to go to a specialist at the GatosMedical hospital for a formal diagnosis. Since GatosMedical falls under the same CaliShield RHIO, the specialist at GatosMedical can access her nasal allergy documents using the Information Management System used by CaliShield. Three years pass by and Sandra relocates to Arizona for her job. The following summer, she gets admitted to the emergency ward of SnakeEyesMedical hospital in the ArizonaCare RHIO after exhibiting acute shortness of breath. The emergency doctor from SnakeEyesMedical requests access to Sandra's allergy treatment documents from CaliShield in order to provide care. At this point, both RHIOs face an interesting dilemma. Does CaliShield provide ArizonaCare

with the information? If so, what conditions apply? Can ArizonaCare pass this information on to some other RHIO? The transaction initiated by SnakeEyesMedical requires compliance with the policies specified by both California and Arizona law, as well as the guidelines for each RHIO.

Further, the emergency medical professional at SnakeEyesMedical may have new observations after treating Sandra. This new information could be very helpful for the practitioners at GoodSam to deliver better, more informed care to Sandra when or if she gets back. When such an addition to Sandra's EMR occurs there may be an obligation on ArizonaCare to notify CaliShield of these changes. Additional complications occur when Sandra files for a permanent move from California to Arizona, in which case the majority of her medical records may have to be transferred to the ArizonaCare RHIO. However, if this happens and Sandra is on a vacation and attends an emergency department in Pennsylvania, the PennCross RHIO in Pennsylvania should be allowed access to all the medical records corresponding to Sandra from ArizonaCare. Now, the question arises whether ArizonaCare has the authority to pass on the documents originated from CaliShield to PennCross.

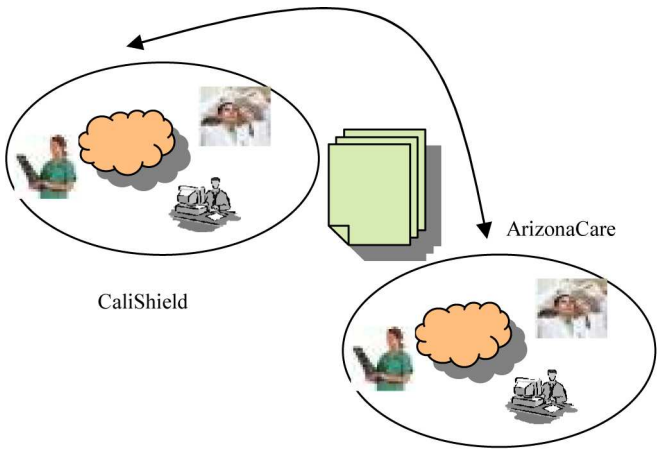


Fig. 6.1: Data Sharing across RHIOs.

6.2 Related Work

The earliest references to work on distributed privacy policy enforcement can be found in the Enterprise Privacy Authorization Language (EPAL) Stufflebeam et al. (2004) specification. Here, the focus is on auditing and establishing trust for a single disclosure object using a single policy. Rivest & Lampson (1996) defines a model in which a recipient is either granted access to the entire document, or must request authorization from the source. The Hippocratic Database (HDB) Agrawal et al. (2002) contains functionality that provides data-level privacy controls to enable disclosure compliance. HDB Active Enforcement (AE) LeFevre et al. (2004) Agrawal et al. (2007) provides cell-level, policy-based disclosure management functionality, such that query execution is compliant with predefined privacy policies. The idea is to rewrite queries to enforce fine grained access control at query execution time. The policies contain rules specifying (i) who is allowed to access what information; and (ii) for what purposes information may be disclosed to each recipient. Policies are expressed in a specification language, such as P3P or EPAL. Each policy contains a number of rules with a description of the condition(s) under which a data column will be disclosed. For a given rule, the (schema, table, column) fields specify the column that the rule applies to. The (purpose, accessor, recipient) triple refers to the purpose of the query, the initiator of the query, and the recipient of the query results respectively. Additionally, the user can define conditions that can be appended to a rule (e.g. the age attribute of patient must be > 18).

6.3 System Design

The problem of distributed enforcement of privacy policies involves multiple challenges:

1. The lack of central authority to ensure that all the nodes involved adhere to the privacy constraints applicable to the shared document(s).

2. Enforcement could possibly involve multiple privacy policies based on the source, destination and the documents involved in the transfer. In the previous example, the system would have to enforce the privacy policies of both CaliShield and ArizonaCare along with Sandra's consent information.

3. Data could be forwarded to a party with additional authorities, such as remote update rights and an authority to forward it to someone else. For example, the medical practitioner at SnakeEyesMedical can update or append prescription information to the document received from CaliShield.

The crux of distributed privacy policy enforcement problem lies in the fact that no single party has *a priori* access to all the policy constraints applicable to a document in a given state of the system. We propose to overcome this problem by identifying all privacy policy constraints applicable to the shared subset(s) of documents and stick them together, forming a single entity of transfer. In taking this approach of packaging policy with data, we maintain centralized decision making in a distributed enforcement. Since we only transfer the policy constraints that apply to the disclosed data, the impact on transfer is relatively small and does not require prior agreement among all participating entities.

6.3.1 Sticky Privacy Policies

We design a *Sticky Privacy Policy* such that each policy consists of three distinct parts: *Data*, *Policy* and *Audit* information. The data section contains part(s) or whole of all the healthcare document(s) associated with the policy. This is followed by the actual policy section which captures the policy constraints applicable to all the documents made available in the data section. The Audit section of the sticky policy consists of the source, requestor, a timestamp and a digital signature to verify the authenticity of the sticky policy. The source and requestor information is used by the auditor to track the data while the timestamp is used to determine the causal ordering. The digital signature serves two purposes: 1) to maintain the integrity of the healthcare documents and guaranteeing that the document(s) are not tampered with and, 2) to ensure the non-repudiation of the sticky policy. Figure 6.2 illustrates this format. The semantics of the policy entries are as follows:

1. **Requestor:** The entity requesting access to the data section from the source. The values for this entry could be taken from the roles mentioned in CDA standard. Some example values are: Emergency Specialist, Medical Resident, Staff Physician etc.

2. **Recipient:** The entity that will be the final consumer of the data. The domain of possible values is similar to the set used for a requestor.

3. **Purpose:** The purpose for which the data is being requested. Possible values include: Preliminary Examination, Administrative, Research etc.

4. **Retention:** The time period until which access to the data is allowed. This could be computed based on various organizational policies.

5. **Copy-forward:** The condition specifying whether the recipient is entitled to forward the requested data to a third party after copying. Based on the healthcare scenario described in Sections 1 and 2 we define a set of possible values for this condition as listed in Table 6.1.

6. **Append/Modify:** The Boolean condition specifying whether the recipient can append/modify the document(s). The possible values are list in Table 6.2.

Copy forward	
Yes w/notification	*May copy and forward the data with a notification to the sender*
Yes w/o notification	*May copy and forward the data without any notification*
No	*May not forward at all*
Ask	*Must ask the sender on forward*

Table 6.1: Copy Forward

6.3.2 Enforcement Model

To address the distributed policy enforcement problem, we assume a trust model where authenticated users are non-malicious and auditable. Our threat model focuses on the occasional curious user who inspects the data they receive and attempts to gather data that they are not entitled to. This does not address malicious users who attempt to gain access to data they have not received.

Our enforcement model consists of two techniques:

Proactive: Proactive enforcement involves prevention of unauthorized disclosure before it occurs by blocking operations or suppressing results that may lead to a violation of policy.

Append/Modify	
Yes w/notification	*May append/modify with a notification*
Yes w/o notification	*May append/modify without any notification*
No	*May not append/modify at all*
Ask	*Must ask on append/modify attempt*

Table 6.2: Append/Modify

Reactive: Reactive enforcement involves detection of the violation through audits. This is based on an optimistic assumption that the environment is non-malicious.

The primary difference between these approaches is when the enforcement occurs. The goal of proactive enforcement is to eliminate all violations before they occur. This approach, on its own, has limited applicability when *a priori* knowledge of all the possible access scenarios is not possible. In the healthcare domain, an emergency case might require access to otherwise sensitive documents to unprivileged users. In order not to impede delivery of care, this violation of the current system policy should be allowed, as long as it is auditable. Reactive enforcement achieves this by tracking all the access information and assuming the existence of a trusted auditor system. The auditor must be able to access data from the source and recipient including any intermediaries between those parties. This could be the responsibility of a central trusted organization such as the US Department of Health and Human Services or its delegates. In the case of a violation, the system presumes the node to be guilty and demands records to certify innocence.

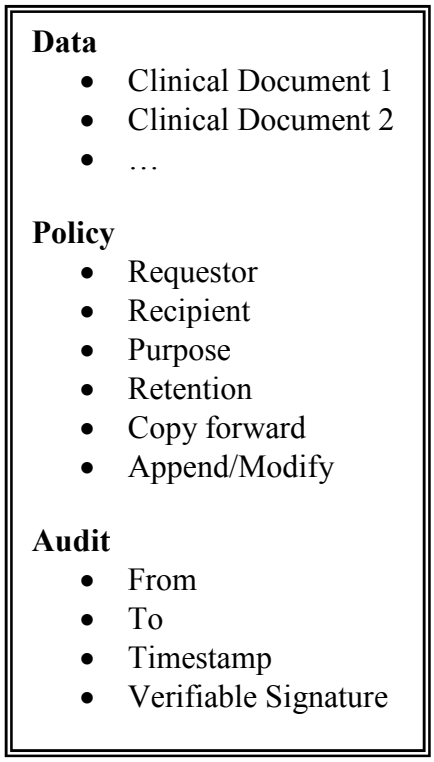


Fig. 6.2: Sticky Privacy Policy Format.

In the case of distributed enforcement, proactive and reactive enforcement can be achieved through either a centralized approach or a federated approach through a set of cooperating enforcers. In our design we achieve proactive enforcement through HDB Active Enforcement technology. In the case of reactive enforcement, a centralized approach employs a single auditor which is trusted and authorized to investigate all aspects of a suspected policy violation. In a decentralized model, a set of cooperating auditors can be employed with each team responsible for a specific set of nodes, data or both. Beyond the issue of proactive and reactive enforcement, there is a question of where the enforcement occurs. We consider two cases: enforcement at the source and enforcement at the recipient. Enforcement at the source is simpler in that it relies on controls at the source. Enforcement at the recipient places trust in all recipients. However,

some enforcement can only occur at the recipient. For example, restricting the recipient from forwarding the result on to others is not something the source can enforce. Our approach is to attempt to perform all enforcement at the source and only rely on enforcement at the recipient when no alternative exists.

6.4 System Architecture

Based on the sticky privacy policy design and the enforcement model presented above, we build a prototype system to demonstrate the functionalities. Figure 6.3 illustrates the architecture of our system. A request for data is placed by firing a query or a set of queries with a specific request-recipient-purpose triple. On receiving the query, the proactive enforcement module rewrites the query to account for all applicable disclosure policies. The rewritten query is directed to the sticky policy module where the query is further modified to format the result into a sticky policy. A digital signature is computed on the source using a User-Defined Function (UDF) and included in the sticky policy. The final sticky policy is then transferred to the recipient in its entirety. For this prototype, we chose to use XML as the data format for representing the sticky policy for multiple reasons. Although, XML processing as in the case of any text processing involves a lot of overhead, it offers the much needed features of platform independence and simplicity. Figure 6.4 provides a sample healthcare sticky policy.

In our prototype, proactive enforcement is achieved by leveraging HDB Active Enforcement. On the recipient, the Active Enforcement component accepts the received sticky policy, assigns a unique id to the policy and stores an unaltered copy of policy for auditing purposes. The policy elements are then decoupled and the corresponding data and policy constraints are extracted. The policy rules are then entered into the *Policy* and *Scope* tables maintained by

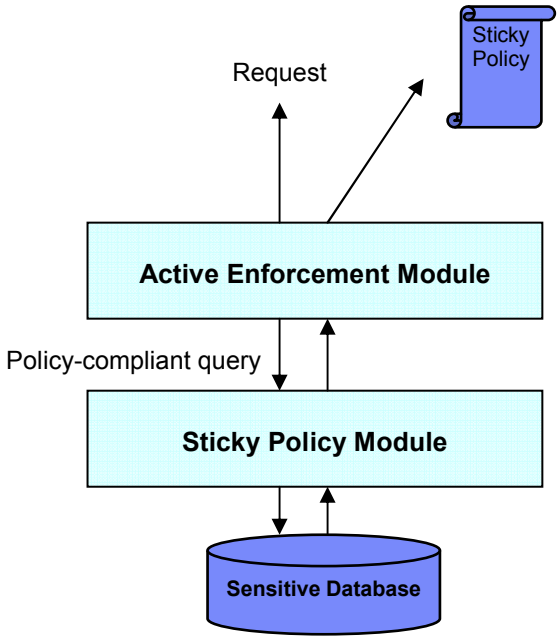


Fig. 6.3: Sticky Policy Enforcement Architecture.

HDB. The document(s) or part(s) of the documents are then stored in a database with an entry in the *Scope* table linking it to the corresponding *Policy* table entries. Figure 6.5 presents the *Generate_Sticky_Policy* and *Consume_Sticky_Policy* algorithms.

Reactive enforcement is achieved through the compliance auditing module by traversing the sticky policy audit logs either to find a violation or to prove innocence. An audit begins with the suspicion of a violation of the privacy policy. We presume that any party with access to the data but without a sticky policy is guilty. In essence, the sticky policy is a certificate of innocence. This is similar to not having a license demonstrating legal ownership of a software product. The auditor searches the database for the data item for which enforcement is presumed to be violated. Once identified, the auditor checks the *Scope* table, and identifies the relevant policy and archive entries. If the ability to identify these entities does not exist (i.e. there is no policy or the archive entry is missing), then a violation has occurred. The auditor then tries to

verify the signature on the sticky policy stored in the archive table. Again if the signature is not valid, a violation has occurred. If everything has been successful so far, the auditor checks to make sure the sticky policy and data content agree. The auditor compares the policy in the policy table to its counterpart in the sticky policy and similarly compares the data in the sticky policy with that in the database. The auditor also verifies that the *Scope* table covers all the data included in the sticky policy. Even if everything checked so far is OK, the audit is not stopped. It is possible for an enforcement breach to have occurred before the current node even received the sticky policy. The auditor then traverses through the sticky policy to identify the node that forwarded the sticky policy. The auditor continues the audit up the chain to the originators until he reaches the bounds of his jurisdiction.

<**ClinicalDocument 1 …**>
<patient><name><given>Phillip</given><family>Barnes</family></name>…
<assignedPerson><name><given>Zachary</given><family>Barnett</family><suffix>M.D.</suffix
></name></assignedPerson>…
<section><title>History of Present Illness</title><text>PhillipBarnes has suffered from calcific bursitis.</text>…
<title>Physical Examination</title>…
<title>Vital Signs</title><text>The patient's height, weight, and body mass index were measured to be 2.29489 meters, 400.05671738536824 pounds, and 34.45 kilograms per meter squared, respectively.</text>…
<**ClinicalDocument 2 …**> **…**

Requestor – Alice/ Admin Role/ ArizonaCare
Recipient - Bob/ Staff Physician/ ArizonaCare
Purpose – Emergency Case
Retention – 30 days
Copy-Forward – Yes With Notification
Append/Modify - No

From – Trina/ Admin Role/ CalShield
To – Alice/ Admin Role/ Arizona Care
Timestamp – Nov 19 2006
Signature - …

Fig. 6.4: Sample Sticky Policy.

6.5 Performance Evaluation

6.5.1 Experimental Setup

We evaluate the performance of our system using a synthetically generated CDA dataset as described in Table 6.3. All experiments were run using IBM DB2 UDB 8.2 on a PC with Pentium-4 2.4GHz processor and a 60GB disk running Microsoft Windows XP with Service Pack 2.

Column	Description
Document ID	Primary Key, Sequential Order
Confidentiality Code	{'N', 'R', 'V'}, Random Order. N – Non Restricted, R – Restricted, V – Very Restricted
Administrative Info	VARCHAR(100)
Billing Info	VARCHAR(100)
Non-Sensitive Clinical Info	VARCHAR(100)
Sensitive Clinical Info	VARCHAR(100)

Table 6.3: Description of the Experimental Dataset Used

6.5.2 Overhead of Sticky Policy Generation

Our first set of experiments measure the overhead incurred in generating the sticky policy. To measure this cost, we consider the total elapsed time to run a sample user query, generate the policy and audit information needed for the sticky policy. This includes all parts of the Generate_Sticky_Policy algorithm except for signature generation which can be ignored as a constant. We use simple selection queries, with policy constraints applying to non-indexed columns. We consider the worst-case scenario, in which the application selectivity is 100%, so that all the cost of privacy processing is incurred and there are no performance gains due to filtering. We

compare the cost of execution of the system with the sticky policy module and the HDB control with the cost of execution of solely the HDB control.

To decide on the number of documents to be included in the sticky policy for evaluating the system there are two possible cases 1) Typical number of medical records stored in a RHIO 2) Typical number of medical records shared across RHIOS. In the first case we can expect the typical number of medical records stored or maintained in a RHIO to be running in the order of millions. However, typical sharing of documents among RHIOS involves querying for documents corresponding to a patient or a set of patients running in the order of hundreds or thousands. We consider this as a more realistic scenario and design our experiments based on this. Figure 6.6 shows the overhead cost of sticky policy generation for tables containing 1000, 2000, 4000 and 8000 documents. It can be observed that the overall cost introduced by sticky policy generation over the privacy preserving query processing in HDB is acceptable considering that the generation is done using XML. This can be improved for performance critical systems through the use of a proprietary format for sticky policies as opposed to XML.

6.5.3 Overhead of Sticky Policy Consumption

We now measure the performance overhead incurred for sticky policy consumption. The goal is to measure the cost involved in accepting a sticky policy and updating the metadata tables with the data. Here again, we measure the elapsed time to consume sticky policies with different data sizes containing 1000, 2000, 4000 and 8000 documents.

Figure 6.7 shows that the bulk of the processing cost deals with XML processing to parse the sticky policy. The time elapsed in updating the metadata tables and the archive is less than 30% of the overall policy consumption cost. As pointed earlier, a proprietary sticky policy format

or specialized XML appliances can be used to reduce the XML processing cost and thereby the overall consumption cost.

6.6 Conclusions

We demonstrated a distributed privacy policy enforcement system using data-level sticky privacy policies. Specifically, we presented the following:

- A system for enforcing privacy policies in a distributed manner, in order to ensure that all privacy regulations corresponding to the data are applied irrespective of who is requesting the data from whom and who is eventually getting access.
- A privacy policy design that allows the user to specify centralized as well as distributed privacy constraints.
- An audit mechanism to detect distributed privacy breaches while allowing necessary exceptions to policy enforcement.

Our design has several limitations. Firstly, we assume that the transmission channel between the source and recipient is secure. Thus, there is little concern placed on the exposure of the documents while in transit. We also assume an agreement on vocabulary among inter-operating parties. Future work should also consider hostile environments and touch on the difficulties in considering Byzantine failures or collusion among participants.

Generate_Sticky_Policy
Input: Query *Q*
Output: Sticky Policy *SP*
Method: Execute the query *Q* and include the result *R* in the sticky policy. For each column c referenced in the "select" section of query *Q*, retrieve all the policies registered in the scope table. Finally, generate the signature and include in the sticky policy *SP*.

R = Execute *Q*
Append(*SP*, *R*)
For all columns *c* referenced by *Q* do
C[] = column list *c*
for i=0 to length of *C* do
Append(*SP*, **GetPolicy**(Request, *Purpose*, *Recipient*, *C[i]*))
Audit= *From*, *To* and *Timestamp*
Append(*SP*, Audit)
Generate Signature
Append(*SP*, Signature)

Consume_Sticky_Policy
Input: Sticky Policy *SP*
Output: Data table and Updated metadata tables
Method: Store unaltered copy of *SP* in its entirety in the archive for audit purposes. Take out the data contents and populate a new table t. Include all the policy constraints listed in the stickypolicy into the policy table and make entries linking to these in the scope table.

Copy(*SP*, archive)
Extract(*SP*, data)
Create table *T*
Insert data values from *SP* into *T*
Extract(*SP*, policy constraints)
Insert policy constraints into *POLICY* table
Insert entries linking *T* and *POLICY* table into *SCOPE* table

Fig. 6.5: Algorithms for sticky policy generation and consumption.

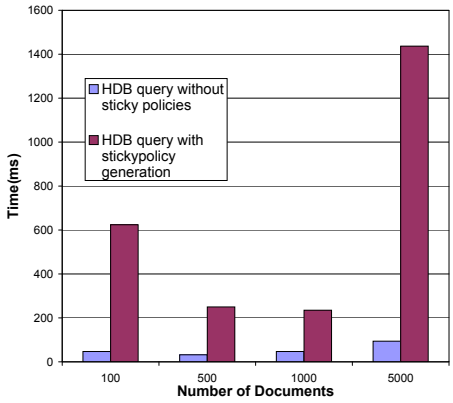


Fig. 6.6: Overhead of Sticky Policy Generation.

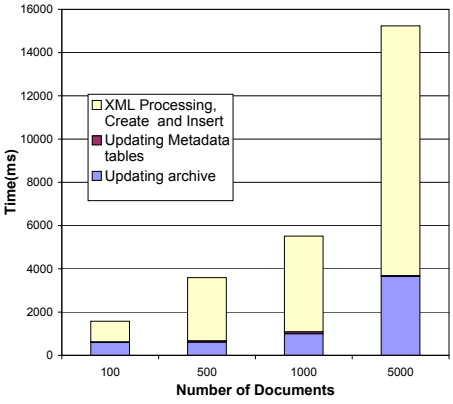


Fig. 6.7: Overhead of Sticky Policy Consumption.

Chapter 7

Information Fusion capable Healthcare Data Warehouse

In this dissertation, we have so far focused on privacy preserving healthcare data management. Apart from data privacy, the other topics that carry a lot of significance in this domain are data availability and knowledge discovery. Healthcare data is composed of a variety of information such as patient data which includes patient demographics and treatment history, clinical data such as tissue and pathology and genomic data such as DNA sequences and gene expressions etc. However, these data sets are managed by disparate organizations such as hospitals, clinical labs, research centers and government agencies that are controlled independently. This hinders the availability of this valuable data to researchers, who end up studying only islands of data. This scenario poses a serious challenge for a global study of the disease and brings out the need for platforms that offer single-point access to patient, clinical, and genomic data from multiple sources. Further, data mining and knowledge discovery on these datasets leads to progress in disease diagnosis, treatment and drug discovery. However, because of the lack of data availability it is typically carried out exclusively on each of the datasets. This leads to limited knowledge discovery as the data does not capture intrinsic relationships among these inter-related sources. For example, biomarker studies are one of the key knowledge discovery tasks involved in disease research that identifies genes that are responsible for the disease. This is traditionally achieved by running cluster analysis on gene expression data alone. Recent advancement in this area Holmes & Bruno (2000) Kasturi & Acharya (2004) shows that inclusion

of clinical and other related data in such studies leads to better results. This brings out the need for platforms that offer tools to mine and explore across heterogeneous healthcare data.

In this chapter, we present the FUZEBASE system developed as part of The Pennsylvania Cancer Alliance for Bioinformatics Consortium (PCABC). The system is aimed at providing the following functionalities:

- A data warehouse for all the research centers that are part of the PCABC consortium.
- A system providing single-point access for patient, clinical and genomic data. A web based interface to the system to make the data available publicly.
- A suite of tools to perform interactive data exploration and information fusion based data mining.

Figure 7.1 gives an overall view of the goal. Section 7.1 provides a description of the data made available through the warehouse and our system architecture. Section 7.2 motivates and demonstrates the suite of tools available on the platform and Section 7.3 provides conclusions.

7.1 Platform

7.1.1 Data

PCABC is a consortium of leading cancer research centers in the state of Pennsylvania: University of Pittsburgh, Fox Chase Cancer Center, the University of Pennsylvania, Thomas Jefferson University, Penn State University and the Wistar Research Institute. The data consists of patient, clinical and genomic information on prostate, breast and melanoma cancer cases studied in the participating research centers and hospitals. The patient data consists of demographics

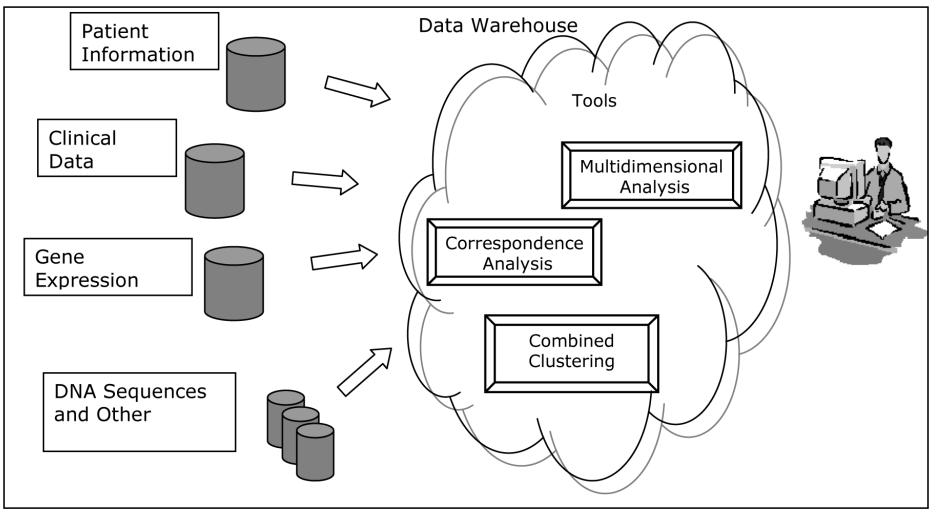


Fig. 7.1: System Overview.

information collected from the patients at the diagnosis stage. This includes information such as patient's age, gender, family history of cancer, previous history of diagnosis, smoking preferences etc. The clinical information consists of around $57,000$ tissue samples collected from all three forms of cancers at all stages of development. The genomic data consists of experimental data such as gene expression data, sequence data and proteomics data.

7.1.2 Architecture

Our system architecture is presented in Figure 7.2. It consists of three layers: 1. The Presentation layer, 2. The Application layer and 3. The Datawarehouse layer. The datawarehouse layer is the core system providing warehouse functionalities including necessary data maintenance, materialized views and multidimensional cubes. It also takes care of the necessary query processing capabilities for integration from heterogeneous data sets stored on the warehouse. The Processor module in this layer accomplishes this by sending pre-processed queries received from the application layer to the splitter which directs them to the appropriate data mart. The

data integrator collects the query results and processes them to cross link related data items and submits the final results back to the processor. The application layer deals with the implementation logic of the tools offered through the system. The results obtained in the application layer are submitted to the presentation layer for visualization and user interpretation.

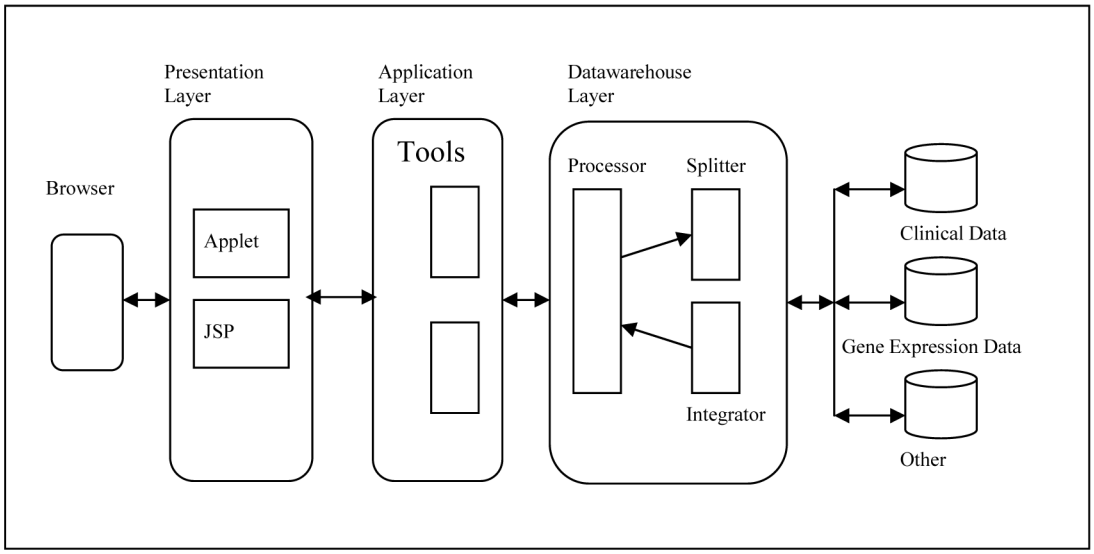


Fig. 7.2: System Architecture.

The key challenge involved in building a datawarehouse is the lack of a common data model. Such a data model should satisfy the following properties:

1. The data model needs to capture multiple inter-related data sets.
2. The model should capture each of the data sets without adding any additional information.
3. The model should capture each of the data sets in a lossless manner.
4. The model should highlight the relationship between any two data sets.
5. The semantics of the model should be easily comprehensible.

In our system, we use an adapted version of the Data Cube model proposed by Gray et al. (1996). The basic idea is to model data as ‘facts’ and ‘dimensions’. The ‘facts’ are the ‘data-of-interest’ and ‘dimensions’ are data with respect to which facts are interpreted. Figure 7.3 shows a conceptual 3-dimensional data cube over patient demographic data. It consists of three dimensions: “Age of the Patient”, “Tissue Sample Type”, and “Race to which the patient belongs to” and a single fact “Number of Patients”. Prior work in Vassiliadis (1998) formalizes data cube based modeling in a similar manner. Tao et al. (2005) applied this model for simultaneous visualization of genotypic and phenotypic data sets.

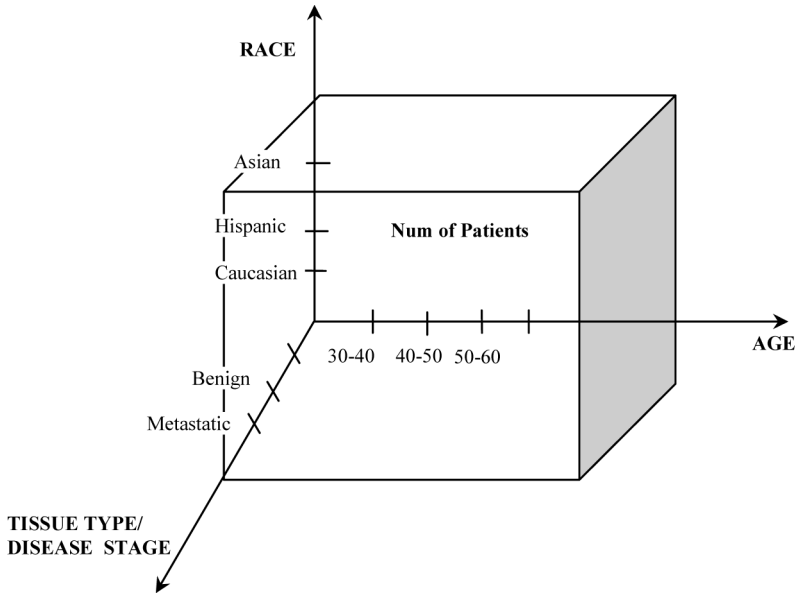


Fig. 7.3: Data Cube.

7.2 Tools

7.2.1 Multidimensional Analysis

Patient data exploration is one of the first steps carried out in disease studies. It provides insights into the spread of the disease in different races and geographic locations. Examples of some simple queries used in such studies are:

1. **What is the average age-at-diagnosis for African American patients?**

2. **What percentage of the patients recorded have a family history of prostate cancer?**

Similarly, such queries can also be posed on clinical, treatment and follow-up and can be extended to span across multiple data sets. Example queries are:

1. **What is the average gene expression vector for patients with Gleason score of 5?**

2. **Based on certain ontology for cellular functions, what is the average expression vector of all the genes corresponding to a particular cellular function?**

Exploratory queries such as these help researchers in identifying disease hot-spots, geographical-spread patterns and other global insights. Thus, there is a need for a tool that allows users to formulate such queries and help them understand the results through data visualization.

The ‘Multidimensional Analysis’ tool provides these functionalities in our system. The tool consists of a query formulation module and a visualization module. The query formulation module helps users formulate exploratory queries and the visualization module helps in the visual interpretation of the result obtained. The user initiates the tool by selecting the subset of the information (e.g., set of patients, disease conditions etc) to be considered for the analysis. Once

this done, the user selects a certain attribute (fact) as the focus of analysis, and some attributes (dimensions) along which the fact values need to be visualized.

For example, consider the query “What percentage of the prostate cancer patients belong to the African-American race and fall in the age range of 50-60?”. In this case, the fact is “Percentage of Patients” and the dimensions are “Race” and “Age-at-diagnosis”. Figure 7.4 depicts the snapshot of the result obtained from the tool by running this query on prostate cancer data available through the warehouse. The utility of the tool can be demonstrated by presenting a sample interpretation of this result: since the graph peaks for the age range 63-67, it can be concluded that maximum percentage of patients are diagnosed with prostate cancer in this age range. The tool also allows users to add dimensions after obtaining a visualization result. Figure 7.5 depicts the snapshot of the result obtained by adding the race of the patient as a dimension to the result obtained earlier in Figure 7.4.

Visualization of the same piece of information at different granularity levels is very useful in human understanding. Our tool facilitates this by allowing the user to perform certain zoom-in and zoom-out operations on the visualization result through user-defined hierarchies. The user is allowed to define these hierarchies on the dimensions selected earlier. It consists of a tree based grouping of all possible values for the dimension. For example, gene ontology can be viewed as a hierarchy defined on genes based on the cellular functions they belong to. Based on these hierarchies, the user can perform the following operations on a visualization result:

Summarize: This allows the user to zoom-out on the result and aggregate the fact values based on dimension hierarchies. Figure 7.6 shows the snapshot of the result obtained by running

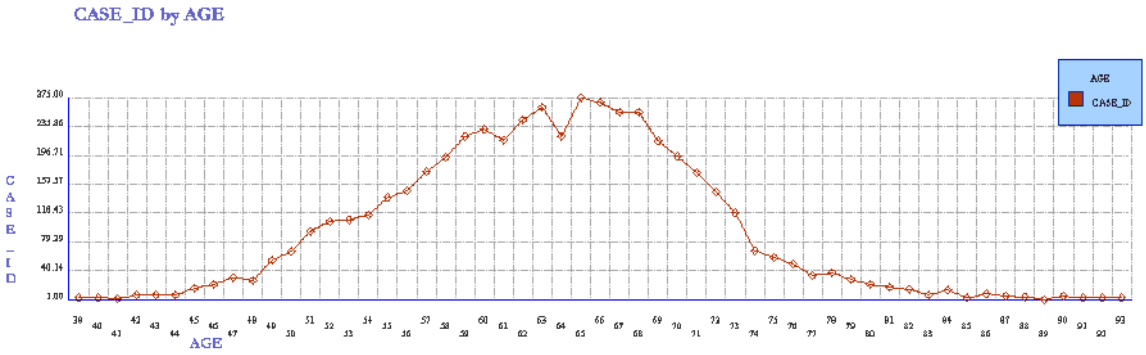


Fig. 7.4: Snapshot of a result from Multidimensional Analysis Tool - Fact: Number of Patients Diagnosed; Dimension: Age of the patient.

the summarize operator on the result obtained in Figure 7.4. We currently support various commonly used aggregation operations such as Average, Maximum, Minimum etc. The user is also allowed to define custom aggregation operators.

Detail: This allows the user to zoom-in on the result by tracking down the dimension hierarchies.

The sample queries dealt with so far involved only patient demographics data. We now demonstrate the functionalities of the tool across multiple data sets. We take clinical data studied in Dhanasekaran et al. (2001) for prostate cancer biomarker identification. We use the corresponding gene expression data available through the warehouse and run some sample queries on this combination of clinical and gene expression data. Figure 7.7 shows the snapshot of the result depicting gene expression values for a set of genes along the clinical stages of the tissues from which the samples are taken. The clinical stages are coded as BPH- Benign, NAP- Normal Adjacent, PCA-Localized and MET- Metastatic. In the original study Dhanasekaran et al. (2001), it was reported that the genes MYBL2 and MYC are over expressed in malignant tumors. Visual interpretation of the result from our tool leads to a similar conclusion and actually goes further and quantifies the amount of over-expression observed in these genes. It can also be observed

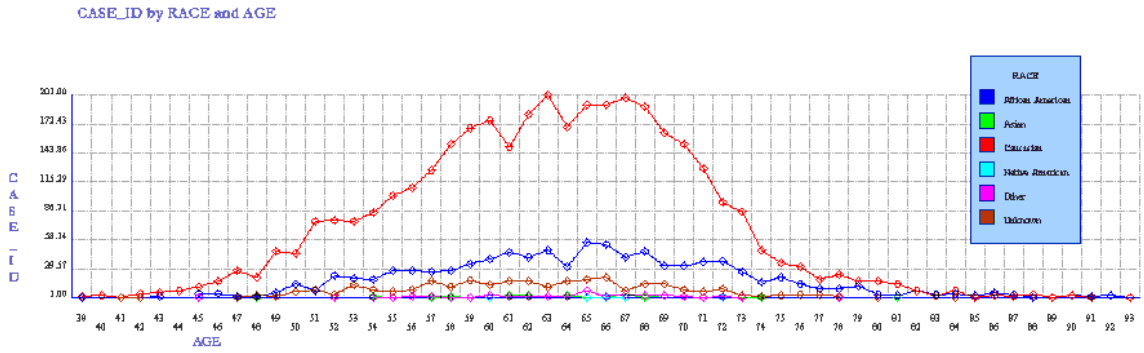


Fig. 7.5: Snapshot of a result from Multidimensional Analysis Tool: Fact: Number of Patients Diagnosed; Dimensions: Age of the patient, Race of the patient.

that MYBL2 is much more dominant in metastatic tumors than when compared to MYC. This observation is supported by running the Summarize operation on the earlier result as shown in Figure 7.8.

7.2.2 Correspondence Analysis

Patient, clinical and genomic data are associated with each other intrinsically. Such associations may lead to trends that correlate parameters in these heterogeneous data sets. For example, clinical data of certain diseases indicate that patients from certain races have lesser age-at-diagnosis when compared to others. So, in this case, the *race* of the patient is related to *average-age-of-diagnosis* of the disease. This may possibly indicate that people of certain races are prone to a disease much earlier than people of other races. Although associations such as the above can be detected by simple queries (finding average age-at-diagnosis for the different races), more complex associations are usually hidden and cannot be detected easily from the data. Thus, there is a need for a tool that helps detect these associations among patient, clinical and genomic data.

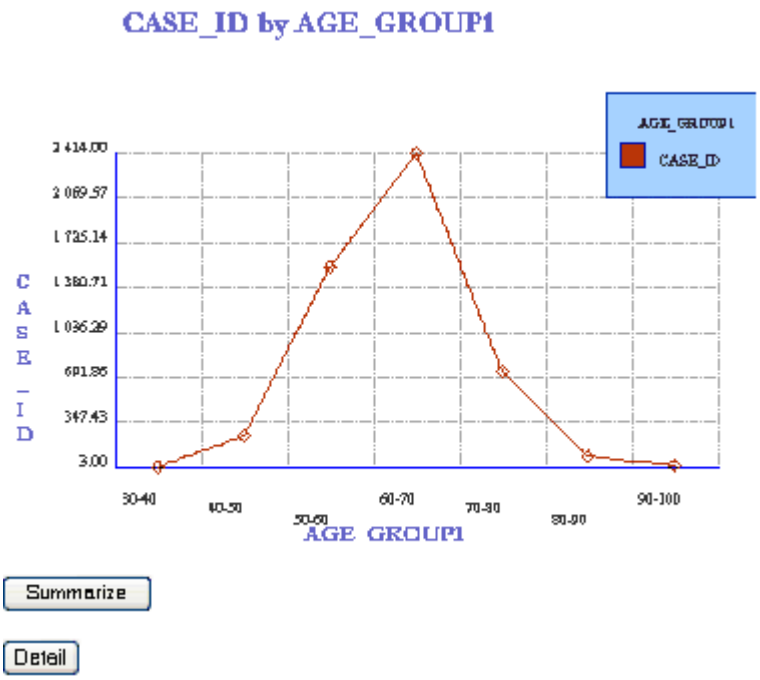


Fig. 7.6: Snapshot of a result from Multidimensional Analysis Tool: Using Summarization on result shown in Figure 7.4.

The Correspondence Analysis tool facilitates this functionality in our system. The tool is based on an exploratory analysis technique proposed in Greenacre (2007). The central idea behind Correspondence Analysis is a *profile*. A profile (either row or column of a data cube) is simply the set of entries normalized by the corresponding total. The output of the tool is a graphical display known as a **map**, a plot in which the row and column profiles are depicted as points. However, unlike some of the other methods such as Principal Component Analysis (PCA) which depicts a low-dimensional projection, Correspondence Analysis displays both the row and column profiles simultaneously on the same plot. Another difference between PCA and Correspondence Analysis is the distance measure used. In PCA, the Euclidean distance measure is used whereas in Correspondence Analysis the $\chi 2$ distance measure is used. We skip further details here for the sake of simplicity and the interested reader is referred to Greenacre (2007)

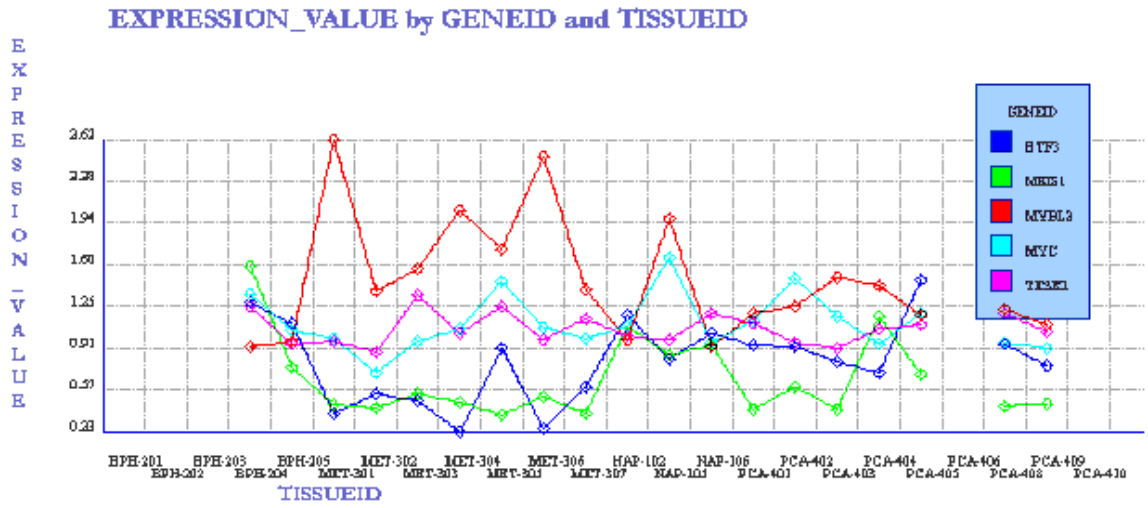


Fig. 7.7: Snapshot of a result from Multdimensional Analysis Tool: Using clinical and gene expression data.

for a thorough exposition of Correspondence Analysis. The output of the tool can be interpreted as follows:

1. The origin corresponds to the centroid of the data profile i.e. the *average profile.*
2. It is comprised of the “optimal displays” of the row and column profiles, although strictly speaking these two sets of points occupy different spaces.
3. The map is scaled such that the row and column points are equally spread out along each principal axis (for a 2D plot, along the horizontal and vertical directions).
4. Although there is no direct interpretation of the distance between a row and a column point, there is certainly a joint interpretation of the row and column points with respect to the principal axes of the map.

We now demonstrate the functionalities of the tool. Consider a 2D data cube along the dimensions “Race” and “Age” of the patient with “total number of patients” as the fact.

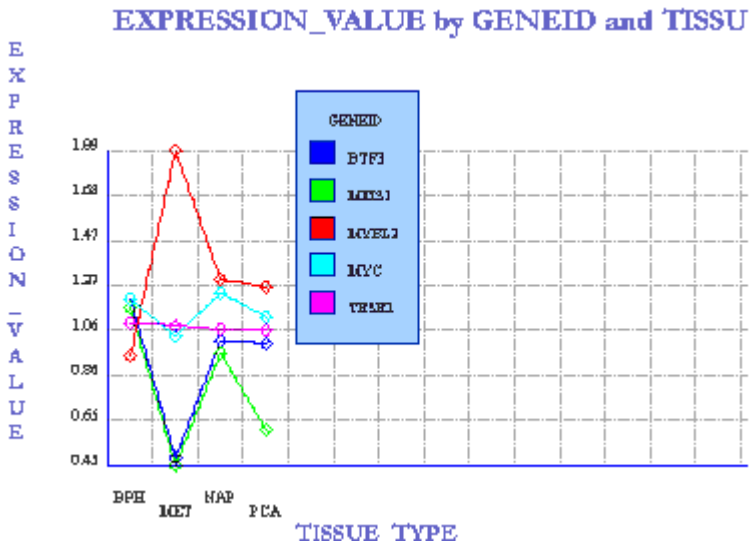


Fig. 7.8: Snapshot of a result from Multdimensional Analysis Tool: Running summarize operation on result shown in Figure 7.7.

Figure 7.9 depicts the snapshot of the result obtained by running the correspondence analysis tool on this dataset. Note that using the earlier description of the plot, one can easily make the following observations from this result.

1. The proximity of the points Age Group “(50-60)” and Race group “Caucasian” to the origin indicates a relatively strong association between them and the average profile. We may hypothesize that middle aged and older Caucasians are more prone to prostate cancer as compared to other age groups.

2. The age groups (<40) and (80-90) which are far away from the origin, are atypical profiles.

Similar interpretations can be made across multiple data sets such as related clinical and gene expression data. Consider the data set Dhanasekaran et al. (2001) used earlier for demonstrating the Multidimensional Analysis tool. The data set contains gene expression profiles of

9984 human cDNA taken from benign and malignant prostate tissues. The tissues are categorized as BPH : Benign Prostate Hyperlysia; NAP : Normal Adjacent Prostate; PCA : Localized and MET : Metastatic. Figure 7.10 shows the snapshot of the result obtained by running the correspondence tool on this dataset. It can be observed that the benign and normal samples (BPH & NAP) lie to the right of the origin (w.r.t. the first principal axis) while the malignant states (PCA and MET samples) lie on the other side. Thus, one can see a separation between the benign and the malignant states which was observed by the original paper through clustering. The genes which lie to the left of the origin tend to have a ”positive association” with the malignant states. Such genes may be candidates for biomarkers especially if they are close to any of the clinical sample points in the graph. Ex. Hepsin, LIM(Enigma),PIM1,MYC were identified in the original paper. These conclusions were actually hypothesized and verified in the original study Dhanasekaran et al. (2001) and the correspondence tool helps the user visualize and interpret them in a comprehensible way.

7.2.3 Combined Clustering

Biomarker studies are one of the key knowledge discovery tasks involved in disease research. In these studies, cluster analysis is run on gene expression data to identify genes with similar gene expression patterns. Recent studies Holmes & Bruno (2000) Kasturi et al. (2003) have shown that including other kinds of information sources such as sequences, ontologies in gene clustering could lead to better results. The Combined Clustering tool provides this functionality by allowing the user to cluster disparate data sets using an information fusion based algorithm developed in Kasturi et al. (2003). This method uses a Self-Organizing Map Kohonen (1995) based clustering algorithm to identify clusters of genes simultaneously based on their

similarity in expression as well as other information sources. Each dataset used is considered a *category*, like expression data, sequence information in the form of DNA motifs, location information, and even gene ontology information. The algorithm is also capable of weighting the data sources, if needed, in order to produce clusters with greater similarity of genes within one data source when compared to the other data sources.

The algorithm uses an iterative procedure by which the probability distribution of the data is reproduced as closely as possible. At each iteration step, a category is randomly selected based on the weighting scheme P. The chosen category r and its associated distance function *dr* are used to train the network of neurons. The weights for the entire input tuple (of dimension $N_1+N_2+\ldots+N_m$) are updated using the Kohonen learning rule Kohonen (1995), although distances are calculated on each segment of the input vector independently using the appropriate distance. Information-theoretic similarity measures such as Kullback-Liebler are preferred for clustering of intensity values Kasturi et al. (2003). To measure similarity between genes based on frequency of motif occurrence, we use a measure based on the Extended Jaccard Similarity coefficient. Interested readers are referred to Kasturi et al. (2003) for further details on the algorithm.

We now demonstrate the usage of the tool and some sample results obtained by running the tool on gene expression data collected from Spellman et al. (1998). The data set consists of gene expression data for yeast cell cycle data. The data set was chosen since it was well studied in the literature and several interesting observations were made based on cluster analysis. The usage of the tool involves the selection of data sets to be clustered, and the weights to be assigned to each data set. The tool outputs the result in a visualizable format depicting the resulting clusters. Figure 7.11(a) and (b) shows the snapshot of the result obtained by running

the combined clustering tool on only: 1. Gene Expression data 2. DNA sequence data (motif frequency data) respectively. Figure 7.11(c) shows the snapshot of the result obtained by running the tool on both the gene expression data and sequence data. The results obtained by using both the data sets show tighter clusters and correlate well with both sequence data and gene expression data Kasturi et al. (2003). Further analysis of one of the clusters obtained using combined clustering showed that 4 out of 5 genes in the cluster, namely, CTF4, POL30, HYS2 and POL32 were mentioned in the original paper to be DNA Syn related genes. These genes also share a common transcription factory MCBa. These results suggest that genes that have a similar function might share a common expression profile and also a common motif.

7.3 Conclusion

The FUZEBASE system is available on the web through the URL http://biogeowarehouse.cse.psu.edu. The consortium members form the primary user-base for the system. However, access is not restricted and the system is available publicly for researchers. The system has been in use since 2004. There are several limitations in the current implementation. Data uploading by the users is restricted to recognized formats. Data cannot be downloaded even if it is de-identified information. Future work on the system includes further development of fusion based knowledge discovery tools and addition of other disease related data.

This work has been published in the 2005 IEEE Computer Based Medical Systems (CBMS) Ganta et al. (2005a), 2005 International Conference on Intelligent Systems for Molecular Biology (ISMB) Ganta et al. (2005b) and 2007 Machine Learning in Bioinformatics book Ganta et al. (2007).

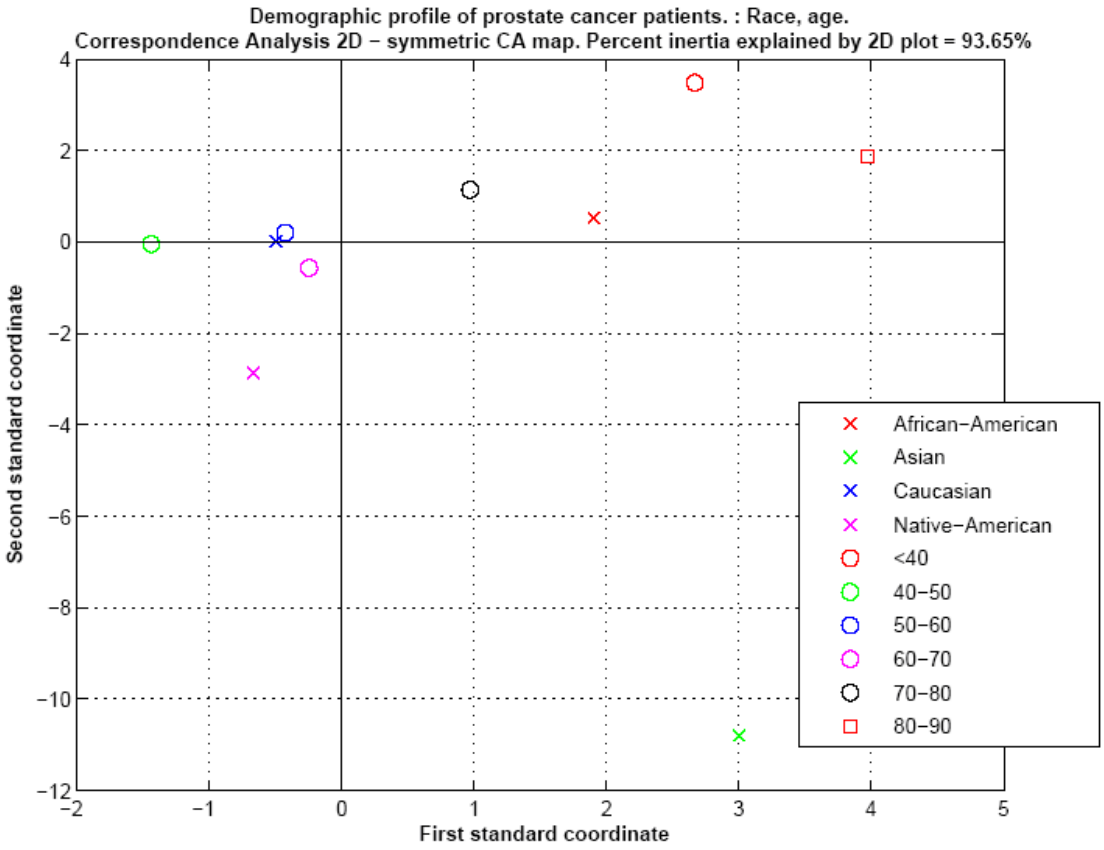


Fig. 7.9: Snapshot of result from Correspondence Analysis Tool: On Patient Data.

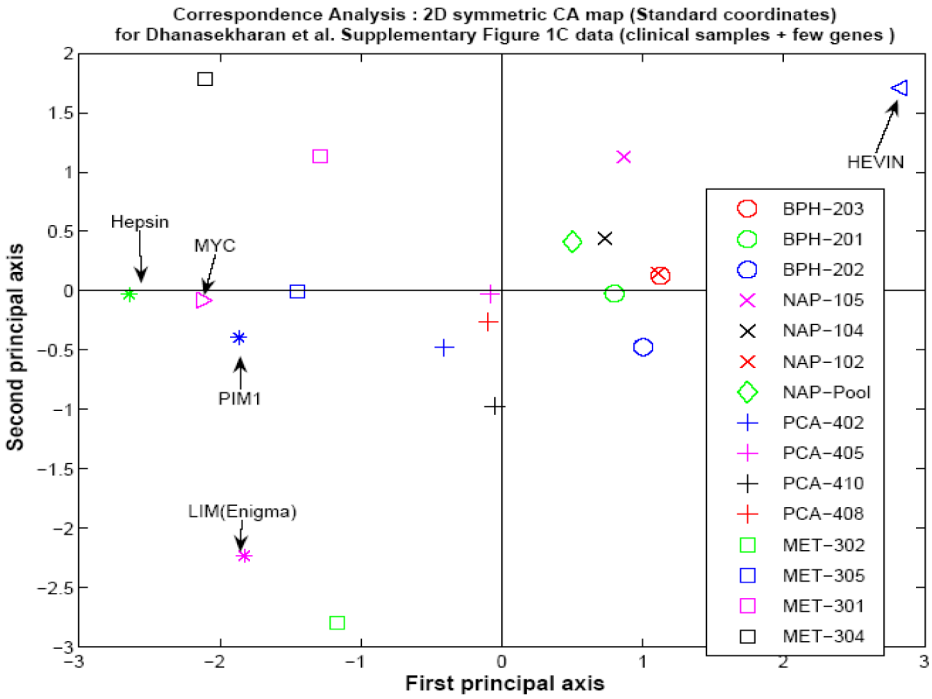


Fig. 7.10: Snapshot of a result from Correspondence Analysis Tool: On Gene Expression Data.

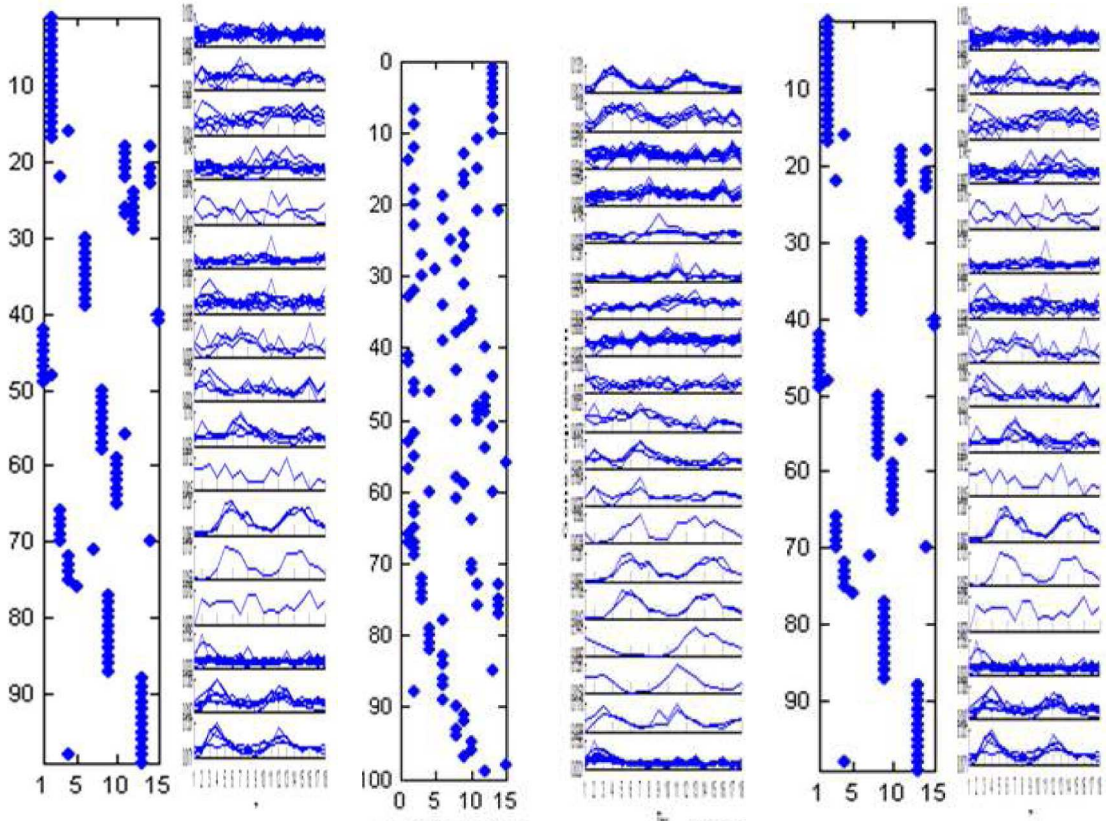


Fig. 7.11: Snapshot of a result from Combined Clustering Tool (a) Clustering based on only Sequence Data (b) Clustering based on only Gene Expression Data (c) Clustering based on both Gene Expression and Sequence Data.

Chapter 8

Conclusions and Future Research

This dissertation set out to study two aspects of healthcare data management: 1. Privacy, and 2. Warehousing. In data privacy, we focused on the data publishing and data sharing scenarios. Investigating privacy preserving data publishing, we modeled two fusion based privacy attacks and studied the effects of these attacks using real world data. In the data sharing scenario, we developed a system to address the problem of cross-enterprise policy-regulated data sharing. In the data warehousing part, we developed a healthcare data warehouse system and information fusion toolkit for medical informatics.

8.1 Summary of Conclusions

8.1.1 Privacy Preserving Data publishing

In Chapter 3, our experimental study proved that several currently proposed partitioning-based anonymization schemes, including k-anonymity and its variants, are vulnerable to independent release attacks. For two different implementations of k-anonymity, we found that sensitive information of a significant percentage of population could be compromised. These methods seem to mitigate independent release attacks to some extent by producing artificially large clusters. Refining the algorithms to produce finer clusters would not help as it will only increase the severity of the independent release attack. The extended definitions of ℓ-diversity

and t-closeness fare better than the original k-anonymity definition but still lead to considerable breach. Additionally, these schemes lead to huge partition sizes and thus result in heavy information loss. Our results indicate that the severity of the attack increases with the increase in entropy of sensitive attribute domain. Further, the severity of the attack increases with the available number of independent releases.

In Chapter 4, our experimental study proved that partitioning-based anonymization schemes such as k-anonymity are vulnerable to web based information fusion attacks. We found that sensitive information can be inferred with significant levels of precision in case of such attacks. While it is not possible to entirely prevent fusion based privacy attacks, one can minimize the extent of breach possible through intelligent data anonymization. Our problem formulation achieves this goal by incremental data anonymization until a threshold value of privacy is attained. An avenue for future work would be to study sophisticated information fusion methodologies to detect inferential attribute disclosure. It would be interesting to look at specific sources on the web that could help an adversary in finding personal information and carrying out such attacks.

Through the work on privacy preserving data publishing, we shed light on the shortcomings of existing solutions in this domain. The adversarial attacks studied prove that a large class of current anonymization schemes are vulnerable in certain scenarios. Our experimental results show the practicality with which such attacks can be carried out and the extent of breach possible. Several questions arise out of this study: Is it possible to find a taxonomy of basic attacks that would inform the development of new anonymization schemes? What should be the main pieces of such a taxonomy? The most important: Is it even possible for partitioning

based anonymization schemes to resist auxiliary information? It would be interesting to see if randomization based anonymization schemes offer satisfactory answers to the same question.

8.1.2 Privacy Preserving Data sharing

In Chapter 6, we demonstrated a distributed privacy policy enforcement system using data-level sticky privacy policies. The main contribution here is a privacy policy design that allows the user to specify centralized as well as distributed privacy constraints. The system enforces all privacy regulations corresponding to the data irrespective of who is requesting the data from whom and who is eventually getting access to it. The system also features an audit mechanism to detect distributed privacy breaches while allowing necessary exceptions to policy enforcement.

Through the work on privacy preserving data sharing, we shed light on the problem of cross-RHIO document sharing. We identified the properties of the existing healthcare system that makes this problem hard to solve. Our system addresses this problem through a data-level sticky privacy policy based system. Several questions come out of this study: Is it possible to provide an ideal data sharing mechanism that allows seamless sharing of policy-regulated data across enterprises. Under what scenarios is this not possible? If so, what features need to be built into policy specification languages such that any potential conflicts arising during data sharing can be resolved or mitigated?

8.1.3 Data Warehousing

In Chapter 7, we presented the FUZEBASE datawarehouse system and information toolkit. The goal here was to demonstrate the significance of information fusion based tools

for biomedical informatics research which could take advantage of the nationwide data sharing infrastructures such as grids. The platform serves two purposes: 1. As a data warehouse for various data sets involved in biomedical informatics studies 2. To provide and demonstrate a set of information fusion tools for disease research. The PCABC consortium members form the primary user-base for the system. However, access is not restricted and the system is available publicly for researchers. It would be interesting to look into mechanisms that allow data privacy concerns to be addressed via such platforms rather at the data source. This can potentially retain utility in the data and better serve the intended purpose of such data warehouses.

8.1.4 Future Directions

In the independent release attack, a natural candidate for future investigation is the release of overlapping contingency tables that are often considered in statistics literature. On the other hand it would be interesting to quantify the resistance offered by randomization based schemes in case of such attacks. In case of web based attacks, an avenue for future work would be to study other information fusion methodologies to infer sensitive data. It would be prudent to identify specific public data sources that would help an adversary in deducing personal information and in assisting such attacks. Another direction would be to study other settings where information fusion attacks are realistic and effective. In the cross-enterprise data sharing work, our system design has several limitations. Firstly, we assume that the transmission channel between the source and recipient is secure. Thus, there is little concern placed on the exposure of the documents during transit. We also assume an agreement on vocabulary among interoperating parties. Future work on this should address these limitations. It is also important to consider hostile environments and touch on the difficulties in considering Byzantine failures or

collusion among participants. In the work on FUZEBASE system, there are several limitations in the current implementation. Data uploading by the users is restricted to recognized formats. Data cannot be downloaded even if it is de-identified information. Future work on this should address these limitations. It is also important to add other disease related data and further the development of fusion based knowledge discovery tools.

Appendix

HIPAA Safe Harbor Provision

The second de-identification provision of the HIPAA Privacy Rule (the Safe Harbor) requires the removal of eighteen specific types of information for any person (e.g., patients, doctors, etc.). Specifically, the rule lists the following U.S. Department of Health and Human Services Office for Civil Rights. HIPAA administrative simplification regulation text (2006):

(A) Names;

(B) All geographic subdivisions smaller than a State, including street address, city, county, precinct, zip code, and their equivalent geocodes, except for the initial three digits of zip code if, according to the current publicly available data from the Bureau of the Census: (1) The geographic unit formed by combining all zip codes with the same three initial digits contains more than 20,000 people; (2) The initial three digits of a zip code for all such geographic units containing 20,000 or fewer people is changed to 000.

(C) All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older;

(D) Telephone numbers;

(E) Fax Numbers;

(F) Electronic mail addresses;

(G) Social security numbers;

(H) Medical record numbers;

(I) Health plan beneficiary numbers;

(J) Account numbers;

(K) Certificate/license numbers;

(L) Vehicle identifiers and serial numbers, including license plate numbers;

(M) Device identifiers and serial numbers;

(N) Web Universal Research Locators (URLs);

(O) Internet Protocol (IP) address numbers;

(P) Biometric identifiers, including finger and voice prints;

(Q) Full face photographic images and any comparable images;

(R) Any other unique identifying number, characteristic, or code, except as permitted by paragraph(C) of this section

Bibliography

Adam, N. R. & Wortmann, J. C. (1989) Security-control methods for statistical databases: a comparative study. ACM Computing Surveys 25(4).

Aggarwal, C., Pei, J. & Zhang, B. (2006) On privacy preservation against adversarial data mining. In: SIGKDD.

Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D. & Zhu, A. (2004) k-anonymity: Algorithms and hardness. Tech. rep., Stanford University .

Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D. & Zhu, A. (2005) Anonymizing tables. In: ICDT. pp. 246–258.

Agrawal, R. & Srikant, R. (2000) Privacy-preserving data mining. In: Proc. of the ACM SIGMOD Conference on Management of Data. ACM Press, pp. 439–450.

Agrawal, R., Kiernan, J., Srikant, R. & Xu, Y. (2002) Hippocratic databases. In: 28th Int'l Conf. on Very Large Databases (VLDB), Hong Kong.

Agrawal, R., Bayardo, R., Faloutsos, C., Kiernan, J., Rantzau, R. & Srikant, R. (2004) Auditing compliance with a hippocratic database. In: 30th Int'l Conf. on Very Large Databases (VLDB. Morgan Kaufmann, pp. 516–527.

Agrawal, R., Grandison, T., Johnson, C. & Kiernan, J. (2007) Enabling the 21st century health care information technology revolution. Commun. ACM 50(2):34–42.

Barbaro, M. & Zeller, T. (2006) A face is exposed for AOL searcher no. 4417749. The New York Times .

Bayardo, R. & Aggarwal, R. (2005) Data privacy through optimal k-anonymization. In: ICDE.

Blum, A., Dwork, C., McSherry, F. & Nissim, K. (2005) Practical privacy: The SuLQ framework. In: PODS. ACM Press, pp. 128–138.

Brucker, P. (1978) On the complexity of clustering problems. Lecture Notes in Economics and Math. Systems .

Byun, J.-W., Sohn, Y., Bertino, E. & Li, N. (2006) Secure anonymization for incremental datasets. In: Secure Data Management. pp. 48–63.

Chawla, S., Dwork, C., Mcsherry, F., Smith, A. & Stockmeyer, L. J. (2005) Toward privacy in public databases. In: In Theory of Cryptography Conference. pp. 363–385.

Chen, B.-C., Ramakrishnan, R. & LeFevre, K. (2007) Privacy skyline: Privacy with multidimensional adversarial knowledge. In: VLDB. pp. 770–781.

Cox, L. H. (1980) Suppression methodology and statistical disclosure control. Journal of the American Statistical Association 75(370):377–385.

Cox, L. H. (1982) Solving confidentiality protection problems in tabulations using network optimization: A network model for cell suppression in the u.s. economic censuses. In: International seminar on statistical confidentiality.

Cox, L. H. (1987) New results in disclosure avoidance for tabulations. In International Statistical Institute Proceedings of the 46th Session. .

Cox, L. H. (1995) Network models for complementary cell suppression. Journal of the American Statistical Association .

Cranor, L., Langheinrich, M. & Marchiori, M. (2002a) A p3p privacy preference exchange language 1.0(appel1.0). W3C Working Draft .

Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M. & Reagle, J. (2002b) The platform for privacy peferences(p3p)1.0 specification. W3C Proposed Recommendation .

Dalenius, T. & Reiss, S. P. (1982) Data-swapping: A technique for disclosure control. Journal of Statistical Planning and Inference 6(1):73–85.

Dhanasekaran, S. M., Barrette, T. R., Ghosh, D., Shah, R., Varambally, S., Kurachi, K., Pienta, K. J., Rubin, M. A. & Chinnaiyan, A. M. (2001) Delineation of prognostic biomarkers in prostate cancer. Nature 412:822–826.

Dobra, A. (2002) Statistical tools for disclosure limitation in multiway contingency tables. PhD Thesis, Carnegie Mellon University .

Dobra, A. & Feinberg, S. E. (2000) Assessing the risk of disclosure of confidential categorical data. Bayesian Statistics .

Dobra, A. & Feinberg, S. E. (2003) Bounding entries in multi-way contingency tables given a set of marginal tables. In Foundations of Statistical Inference: Proceedings of Shoresh Conference 2000 .

Domingo-Ferrer, J. & Mateo-Sanz, J. M. (2002) Practical data-oriented microaggregation for statistical disclosure control. IEEE Transactions on Knowledge and Data Engineering 14(1):189–201.

Duncan, G. T. & Feinberg, S. E. (1997) Obtaining information while preserving privacy: A markov perturbation method for tabular data. Joint Statistical Meetings .

Evfimievski, A., Srikant, R., Agrawal, R. & Gehrke, J. (2002) Privacy preserving mining of association rules. In: Proc. of 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD). pp. 217–228.

Fung, B. C. M., Wang, K. & Yu, P. S. (2005) Top-down specialization for information and privacy preservation. In: ICDE '05: Proceedings of the 21st International Conference on Data Engineering. IEEE Computer Society, Washington, DC, USA, pp. 205–216.

Fung, B. C. M., Wang, K., Fu, A. W.-C. & Pei, J. (2008) Anonymity for continuous data publishing. In: EDBT. ACM Press, pp. 264–275.

Ganta, S. R. & Acharya, R. (2008a) Adaptive data anonymization against information fusion based privacy attacks on enterprise data. In: SAC. pp. 1075–1076.

Ganta, S. R. & Acharya, R. (2008b) On breaching enterprise data privacy through adversarial information fusion. In: ICDE Workshops. pp. 246–249.

Ganta, S. R. & Acharya, R. (2008c) On breaching enterprise data privacy through adversarial information fusion. CoRR abs/0801.1715.

Ganta, S. R., Kasturi, J., Gilbertson, J. & Acharya, R. (2005a) An online analysis and information fusion platform for heterogeneous biomedical informatics data. In: CBMS. pp. 153–158.

Ganta, S. R., Kasturi, J., Narasimhamurthy, A. & Acharya, R. (2005b) Fuzebase: An online platform for exploration and information fusion of biomedical informatics data. In: ISMB.

Ganta, S. R., Kasturi, J., Narasimhamurthy, A. & Acharya, R. (2007) An Information Fusion framework for Biomedical Informatics, John Wiley and Sons, chap. 20.

Ganta, S. R., Kasiviswanathan, S. P. & Smith, A. (2008a) Composition attacks and auxiliary information in data privacy. In: KDD. pp. 265–273.

Ganta, S. R., Kasiviswanathan, S. P. & Smith, A. (2008b) Composition attacks and auxiliary information in data privacy. CoRR abs/0803.0032.

Gray, J., Bosworth, A., Layman, A., Pirahesh, H., Microsoft, J. G., Microsoft, A. B. & Microsoft, A. L. (1996) Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. In: 12th International Conference on Data Engineering. pp. 152–159.

Greenacre, M. (2007) Correspondence Analysis in Practice, Second Edition. Chapman & Hall/CRC, Boca Raton, FL. ISBN 1-584-88616-1.

G.T.Duncan (1990) Inferential disclosure -limited microdata dissemination. Proceedings of the Survey Research Methods Section, American Statistical Association .

G.T.Duncan & Lambert (1986) Disclosure limited data dissemination. Journal of American Statistical Association .

Holmes, I. & Bruno, W. J. (2000) Finding regulatory elements using joint likelihoods for sequence and expression profile data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology. AAAI Press, pp. 202–210.

ICD9 (2008) International classification of diseases, http://www.cdc.gov/nchs/about/otheract/icd9/abticd9.htm.

I.P.Fellegi (1972) On the question of statistical confidentiality. Journal of the American Statistical Association .

Iyengar, V. S. (2002) Transforming data to satisfy privacy constraints. In: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, pp. 279–288.

Kasturi, J. & Acharya, R. (2004) Clustering of diverse genomic data using information fusion. In: SAC '04: Proceedings of the 2004 ACM symposium on Applied computing. ACM, New York, NY, USA, pp. 116–120.

Kasturi, J., Acharya, R. & Ramanathan, M. (2003) An information theoretic approach for analyzing temporal patterns of gene expression. Bioinformatics 19(4):449–458.

Kohonen, T. (1995) Self-Organizing Maps. Springer, Berlin.

Kosko, B. (2005) Neural Networks and Fuzzy Systems. Prentice Hall.

LeFevre, K., Agrawal, R., Ercegovac, V., Ramakrishnan, R., Xu, Y. & DeWitt, D. (2004) Limiting disclosure in hippocratic databases. In: VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases. VLDB Endowment, pp. 108–119.

LeFevre, K., DeWitt, D. & Ramakrishnan, R. (2006a) Mondrian multidimensional k-anonymity. In: ICDE. p. 25.

LeFevre, K., DeWitt, D. J. & Ramakrishnan, R. (2006b) Workload-aware anonymization. In: KDD. ACM Press, pp. 277–286.

Li, N., Li, T. & Venkatasubramanian, S. (2007) t-closeness: Privacy beyond k-anonymity and l-diversity. In: ICDE. pp. 106–115.

Machanavajjhala, A., Gehrke, J. & Kifer, D. (2006) l-diversity: Privacy beyond k-anonymity. In: ICDE.

Machanavajjhala, A., Kifer, D., Gehrke, J. & Venkitasubramaniam, M. (2007) l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data 1(1).

Martin, D. J., Kifer, D., Machanavajjhala, A., Gehrke, J. & Halpern, J. Y. (2007) Worst-case background knowledge for privacy-preserving data publishing. In: ICDE. pp. 126–135.

Mateo-Sanz, J. & J.Domingo-Ferrer (1878) A method for data oriented on the complexity of clustering problems. Lecture Notes in Economics and Math. Systems .

McSherry, F. & Talwar, K. (2007) Differential privacy in mechanism design. In: FOCS. IEEE Computer Society, pp. 94–103.

Meyerson, A. & Williams, R. (2004a) On the complexity of optimal k-anonymity. In PODS .

Meyerson, A. & Williams, R. (2004b) On the complexity of optimal k-anonymity. In: PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, New York, NY, USA, pp. 223–228.

Miklau, G. & Suciu, D. (2003) Controlling access to published data using cryptography. In VLDB .

Miklau, G. & Suciu, D. (2004) A formal analysis of information disclosure in data exchange. In SIGMOD .

Narayanan, A. & Shmatikov, V. (2006) How to break anonymity of the netflix prize dataset. CoRR abs/cs/0610105.

Pei, J., Xu, J., Wang, Z., Wang, W. & Wang, K. (2007) Maintaining k-anonymity against incremental updates. In: SSDBM. p. 5.

Policies in focus: Strengthening Healthcare (2006) http://www.whitehouse.gov/infocus/healthcare.

Re-engineering the Clinical Research Enterprise (2005) http://nihroadmap.nih.gov/clinicalresearch/index.asp.

Rivest, R. L. & Lampson, B. (1996) Sdsi - simple distributed security infrastructure. Tech. rep.

Samarati, P. & Sweeney, L. (1998) Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory .

Slavkovic, A. & Feinberg, S. E. (2004) Bounds for cell entries in two-way tables and given conditional frequencies. Privacy in Statistical Databases .

Snodgrass, R. T., Yao, S. S. & Collberg, C. (2004) Tamper detection in audit logs. In: Proceedings of the International Conference on Very Large Databases. pp. 504–515.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D. & Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell 9(12):3273–3297.

Stufflebeam, W., Stufflebeam, W., Antn, A. I., Antn, A. I., He, Q., He, Q., Jain, N. & Jain, N. (2004) Specifying privacy policies with p3p and epal: Lessons learned. In: In Workshop on Privacy in the Electronic Society, WPES-2004. p. 35.

Sweeney, L. (2000) Uniqueness of simple demographics in the u.s population. Tech Rep. Carnegie Mellon University .

Sweeney, L. (2002) k-anonymity: A model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10(5):557–570.

Tao, Y., Friedman, C. & Lussier, Y. A. (2005) Visualizing information across multidimensional post-genomic structured and textual databases. Bioinformatics 21(8):1659–1667.

The California Regional Health Information Organization (2006) http://www.calrhio.org/.

The Goals of Strategic Network (2005) http://www.hhs.gov/healthit/goals.html.

UCI Machine Learning Repository (2008) http://www.ics.uci.edu/ mlearn/databases/.

U.S. Department of Health and Human Services Office for Civil Rights. HIPAA administrative simplification regulation text (2006) .

van den Hout, A. & van der Heijden, P. G. (2002) Randomized response, statistical disclosure control and misclassification: A review. International Statistical Review 70:269–288.

Vassiliadis, P. (1998) Modeling multidimensional databases, cubes and cube operations. In: SSDBM '98: Proceedings of the 10th International Conference on Scientific and Statistical Database Management. IEEE Computer Society, Washington, DC, USA, pp. 53–62.

Wang, K. & Fung, B. C. M. (2006) Anonymizing sequential releases. In: KDD. pp. 414–423.

Wang, K., Yu, P. S. & Chakraborty, S. (2004) Bottom-up generalization: A data mining solution to privacy protection. In: ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining. IEEE Computer Society, Washington, DC, USA, pp. 249–256.

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

Winkler, W. (2002) Using simulated annealing for k-anonymity. Research Report 2002-07, US Census Bureau Statistical Research Division .

Wong, R. C.-W., Fu, A. W.-C., Wang, K. & Pei, J. (2007) Minimality attack in privacy preserving data publishing. In: VLDB. pp. 543–554.

Xiao, X. & Tao, Y. (2006a) Anatomy: Simple and effective privacy preservation. In: VLDB. pp. 139–150.

Xiao, X. & Tao, Y. (2006b) Personalized privacy preservation. In: SIGMOD Conference. pp. 229–240.

Xiao, X. & Tao, Y. (2007) M-invariance: towards privacy preserving re-publication of dynamic datasets. In: SIGMOD Conference. pp. 689–700.

Yang, X. & Li, C. (2004) Secure xml publishing without information leakage in the presence of data inference. In VLDB .

Yao, C., Wang, X. S. & Jajodia, S. (2005) Checking for k-anonymity violation by views. In: VLDB. VLDB Endowment, pp. 910–921.

Yasnoff, W. A., Humphreys, B. L., Overhage, J. M., Detmer, D. E., Brennan, P. F., Morris, R. W., Middleton, B., Bates, D. W. & Fanning, J. P. (2004) A consensus action agenda for achieving the national health information infrastructure .

Vita

Srivatsava Ranjit Ganta received his Bachelor of Technology (B.Tech.) in Computer Science and Engineering from the Indian Institute of Technology, Chennai, India in May 2002. He joined the doctoral program in the department of Computer Science and Engineering at Pennsylvania State University in August 2002. His research interests included Data Privacy, Database Systems, Data Mining, Data Warehousing and Healthcare Informatics. He was advised by Prof. Raj Acharya and Asst Prof. Adam Smith. Throughout his graduate studies, he worked as a teaching assistant for several graduate-level courses including Database Systems (CSE 541), Computer Security (CSE 543) and Knowledge Discovery and Data Mining (CSE 598E). He also worked as a summer research intern at the IBM Almaden Research Center during 2006 and 2007. He graduated with a Ph.D. in Computer Science and Engineering in May 2009.