Foundations and Trends® in
Databases
Vol. 2, Nos. 1–2 (2009) 1–167

DOI: 10.1561/1900000008

now
the essence of knowledge

# Privacy-Preserving Data Publishing

By Bee-Chung Chen, Daniel Kifer, Kristen LeFevre
and Ashwin Machanavajjhala

## Contents

Foundations and Trends® in
Databases
Vol. 2, Nos. 1–2 (2009) 1–167

DOI: 10.1561/1900000008

now
the essence of knowledge

# Privacy-Preserving Data Publishing

**Bee-Chung Chen[1], Daniel Kifer[2], Kristen LeFevre[3] and Ashwin Machanavajjhala[4]**

[1] *Yahoo! Research, USA, beechun@yahoo-inc.com*
[2] *Penn State University, USA, dkifer@cse.psu.edu*
[3] *University of Michigan, USA, klefevre@eecs.umich.edu*
[4] *Yahoo! Research, USA, mvnak@yahoo-inc.com*

## Abstract

Privacy is an important issue when one wants to make use of data that involves individuals' sensitive information. Research on protecting the privacy of individuals and the confidentiality of data has received contributions from many fields, including computer science, statistics, economics, and social science. In this paper, we survey research work in privacy-preserving data publishing. This is an area that attempts to answer the problem of how an organization, such as a hospital, government agency, or insurance company, can release data to the public without violating the confidentiality of personal information. We focus on privacy criteria that provide formal safety guarantees, present algorithms that sanitize data to make it safe for release while preserving useful information, and discuss ways of analyzing the sanitized data. Many challenges still remain. This survey provides a summary of the current state-of-the-art, based on which we expect to see advances in years to come.

# 1

# Introduction

> I have as much privacy as a goldfish in a bowl.
>
> — Princess Margaret

Privacy is an important issue when one wants to make use of data that involve individuals' sensitive information, especially in a time when data collection is becoming easier and sophisticated data mining techniques are becoming more efficient. It is no surprise that research on protecting the privacy of individuals and the confidentiality of data has received many contributions from many fields such as computer science, statistics, economics, and social science. With the current rate of growth in this area it is nearly impossible to organize this entire body of work into a survey paper or even a book. Thus we have proceeded with a more modest goal. This survey describes research in the area of privacy-preserving data publishing. We are mainly concerned with data custodians such as hospitals, government agencies, insurance companies, and other businesses that have data they would like to release to analysts, researchers, and anyone else who wants to use the data. The overall intent is for the data to be used for the public good: in the evaluation of economic models, in the identification of social trends, and in the pursuit of the state-of-the-art in various fields. Usually, such

data contain personal information such as medical records, salaries, and so on, so that a straightforward release of data is not appropriate. One approach to solving this problem is to require data users to sign non-disclosure agreements. This solution will need significant legal resources and enforcement mechanisms and may be a barrier to wide dissemination of the data. Furthermore, this cannot protect against data theft even when the victim takes reasonable precautions. Thus, it is important to explore technological solutions which anonymize the data prior to its release. This is the focus of this survey.

In Section 1, we begin by describing the information-protection practices employed by census bureaus (Section 1.1), and we motivate the importance of considering privacy protection in data publishing through a number of real-world attacks (Section 1.2). We then use a simple example (Section 1.3) to introduce the problem and its challenges (Section 1.4). Section 2 is devoted to formal definitions of privacy, while Section 3 is devoted to ways of measuring the utility of sanitized data or the information lost due to the sanitization process. In Section 4, we present algorithms for sanitizing data. These algorithms seek to output a sanitized version of data that satisfies a privacy definition and has high utility. In Section 5, we discuss how a data user can make use of sanitized data. Then, in Section 6, we discuss how an adversary might attack sanitized data. In Section 7, we cover emerging applications and their associated research problems and discuss difficult problems that are common to many applications of privacy-preserving data publishing and need further research.

Having explained what this survey is about, we will now briefly mention what this survey is not about. Areas such as access control, query auditing, authentication, encryption, interactive query answering, and secure multiparty computation are considered outside the scope of this paper. Thus we do not discuss them except in places where we deem this to be necessary. We also focus more on recent work as many of the older ideas have already been summarized in book and survey form [4, 263, 264]. Unfortunately, we cannot cover every technique in detail and so the choice of presentation will largely reflect the authors' bias. We have tried to cover as much ground as possible and regret any inadvertent omissions of relevant work.

## 1.1 Information Protection in Censuses, Official Statistics

The problem of privacy-preserving data publishing is perhaps most strongly associated with censuses, official processes through which governments systematically collect information about their populations. While emerging applications such as electronic medical records, Web search, online social networks, and GPS devices have heightened concerns with respect to collection and distribution of personal information, censuses have taken place for centuries, and considerable effort has focused on developing privacy-protection mechanisms in this setting. Thus, we find it appropriate to begin this survey by describing some of the diverse privacy-protection practices currently in place at national census bureaus and affiliated statistical agencies around the world.

### 1.1.1 Public-Use Data

Most related to the topic of this survey is the problem of releasing public-use data sets. Worldwide, many (though not all) governmental statistical agencies distribute data to the public [54, 58, 133, 234] to be used, for example, in demographic research. However, it is also a common belief that these public-use data sets should not reveal information about individuals in the population. For example, in the United States, Title 13 of the US Code requires that census information only be collected to produce statistics, and that census employees be sworn to protect confidentiality.

Thus, over the years, government statistical agencies have developed a variety of mechanisms intended to protect individual privacy in public-use data. (This research area is commonly known as *statistical disclosure limitation* or *confidentiality*, and it is a subset of the broader field of *official statistics*.) Historically, this work has focused on two main classes of data that are commonly released by governmental agencies:

- **Aggregate count data (*contingency tables*)** Contingency tables contain frequency count information, tabulated on the basis of one

of more variables.[1] For example, a contingency table might contain a population count based on *Zip Code*, *Age Range*, and *Smoking Status*; i.e., in each zip code and each age range, how many people smoke?

- **Non-aggregate data (*Microdata*)** Microdata are simply conventional (non-aggregate) data, where each row refers to a person in the population.

In order to limit the possibility that an individual could be identified from the public-use data, statistical agencies commonly use a combination of techniques [54, 58, 59, 95, 133, 234, 257]; however, statistical disclosure limitation experts at statistical agencies do not typically provide details of the mechanisms used for confidentiality, only generic descriptions. A recent report [95] outlines, in general terms, the practices of the various federal agencies in the United States. (We will describe some of these techniques in more detail in Section 4.)

- **Cell suppression and noise addition (for contingency tables)** In contingency tables, it is common to suppress cells with small counts (*primary suppression*), as well as additional cells that can be inferred using marginal totals (*complementary suppression*). Similarly, it is common to make small perturbations to the counts.
- **Data swapping (for microdata and contingency tables)** Data swapping is a method of making controlled changes to microdata; modified contingency tables can also be re-computed from the results. This technique was used in the United States during the 1990 and 2000 censuses [101].
- **Sampling, geographic coarsening, and top/bottom-coding (for microdata)** For microdata, it is common to only release a subset of respondents' data (e.g., a 1% sample). In addition, it is common to restrict geographic identifiers to regions containing at least a certain population. (In the United States, this is typically 100,000 [257].) It is also common to "top-code" and "bottom-code" certain values. For example, if there are sufficiently few respondents

[1] In SQL, this is analogous to releasing the answer to a COUNT(*) query with one or more attributes in the GROUP BY clause.

over age 90, then a top-coding approach would replace all ages $\geq 90$ with the value 90.

- **Synthetic data (for microdata)** Finally, sometimes synthetic data are generated. The idea is to produce data with similar distributional characteristics to the original microdata. The US Census Bureau is considering using a synthetic data approach to release microdata following the 2010 census [272].

Many of the above-mentioned mechanisms for microdata and contingency table sanitization, respectively, have been implemented in the $\mu$- and $\tau$- Argus software packages [127, 128]; these packages have also been used extensively by Statistics Netherlands.

The US Census Bureau also provides an online (real-time) system called the American FactFinder Advanced Query System [122], which provides custom tabulations (count queries) from the census data. Disclosure control in this system is done primarily by applying queries to the sanitized (e.g., swapped) microdata, and also by imposing cell suppression and top-coding rules to the results.

### 1.1.2 Restricted-Use Data, Research Data Centers, and Remote Servers

While many statistical agencies release sanitized public-use data sets, there is also a commonly held belief that certain data (e.g., high-precision geographical units) cannot be sanitized enough to release, or that the process would yield the data useless for certain kinds of research. For these reasons, federal agencies in the United States [256, 225], Canada [46], and Germany [219] have also set up secure *research data centers* to allow outside researchers to access more precise and detailed data. The idea is to provide a secure physical facility, staffed by census personnel, in which vetted researchers can carry out approved studies using computers with limited external access. In the United States, there are approximately a dozen such locations. Before conducting a study, a researcher must undergo a background check and provide a sworn statement. Before removing results or data from the center, the results must undergo a strict disclosure review, which is conducted by Census Bureau personnel. Similarly, a variety of countries

provide "virtual" secure research data centers (also known as *remote access servers*) that serve a similar purpose [214].

While secure facilities and data centers are not the topic of this survey, this example highlights the multifaceted nature of the privacy-protection problem. Technical tools for privacy-preserving data publishing are one weapon in a larger arsenal consisting also of legal regulation, more conventional security mechanisms, and the like. In addition, this example highlights a (perceived and sometimes formal) tradeoff between *privacy* and *utility*, a theme that has been repeated throughout the literature and that will be repeated throughout this survey.

## 1.2 Real-World Attacks and Attack Demonstrations

A number of real-world attacks and demonstrations indicate the importance of taking privacy into consideration when publishing personal data. In this section, our goal is to briefly recap some notable recent events and attacks, which serve to illustrate the challenges in developing privacy-preserving publishing tools.

One published attack on (purportedly) de-identified data was described by Sweeney [241]. The dataset in consideration was collected by the Group Insurance Commission (GIC) and contained medical records of Massachusetts state employees. Since the data did not contain identifiers such as names, social security numbers, addresses, or phone numbers, it was considered safe to give the data to researchers. The data did contain demographic information such as birth date, gender, and zip code. Unfortunately, it is not common for two individuals to have the same birth date, less common for them to also live in the same zip code, and less common still for them to also have the same gender. In fact, according to the Massachusetts voter registration list (available at the time for $20), no one else had the same combination of birth date, gender, and zip code as William Weld, who was then the governor. Thus, his medical records were easy to identify in the data provided by GIC. This sort of attack, where external data are combined with an anonymized data set, is called a *linking attack*.

Not all linking attacks are as simple as performing a join between the GIC data and the voter registration list. This is especially true for text.

As an example, consider the case of AOL. On Sunday, August 6, 2006, AOL released a 2 GB file containing approximately 20 million search queries from 650,000 of its users, which were collected over a period of three months [24]. In addition to the queries themselves, the data set contained information such as which URL from the search results was clicked and what was its ranking. Although the data set was withdrawn within a few hours, it had already been widely downloaded. The anonymization scheme used to protect the data consisted of assigning a random number (pseudonym) to each AOL user and replacing the user id with this number. Three days later, two New York Times reporters [28] found and interviewed user number 4417749 from the data set. They tracked down this user based on the semantic information contained in her search queries: the name of a town, several searches with a particular last name, age-related information, etc. In the case of AOL, there was no single authoritative table (such as a voter list) to link against; instead, there were many scattered sources of information that were used. The privacy breach occurred since AOL failed to reason about these sources and about the semantic content of search queries. We will return to a more detailed discussion of state-of-the-art privacy protection tools for search logs in Section 7.2.

A few months later, Netflix, a movie rental service, announced the Netflix Prize for the development of an accurate movie recommendation algorithm. To aid participants in their research efforts, Netflix also released a data set of 100 million ratings for 18,000 movie titles collected from 480,000 randomly chosen users. Personal information had been removed, and user ids were replaced with pseudonyms, as in the AOL data. This data set contained movie ratings and the dates when the ratings were created [191]. The high-dimensionality of the data set proved to be a tempting target and an attack on such a data set was anticipated by Frankowski et al. [105], who showed that movie ratings can be linked to posts in an online forum. The Netflix data were attacked shortly after it came out by Narayanan and Shmatikov [186], who showed that external information (such as IMDB reviews) can indeed be linked to the Netflix data set using techniques that are commonly known as *record linkage*. Record linkage was first formalized in the 1960s by Fellegi and Sunter [96]; for a survey, see [270]. Record linkage techniques are

frequently used to estimate *re-identification probabilities*: the probabilities that users in a data set can be re-identified through auxiliary data [268]. These techniques can often handle varying amounts of noise in the auxiliary data, and are also commonly used for the purpose of data cleaning.

Finally, even further illustrating the vulnerability of public personal data sets, several recent attacks have been demonstrated on (purportedly) de-identified social network graphs. Social networks describe a set of people (nodes) and the relationships between them (edges). As in the cases of search logs and movies, a graph can be considered naively anonymized if all identifying characteristics of the people (e.g., names, etc.) have been removed and replaced with pseudonyms. Interestingly, though by this point perhaps unsurprising, a series of attacks have illustrated the fallacy of this approach. Using data from LiveJournal (a blogging site), Backstrom et al. [26] demonstrated that it is often possible for a particular user to re-identify himself in a social network graph, and with minimal collusion, he can frequently re-identify a large fraction of users. Hay et al. [123] and Narayanan and Shmatikov [187] both took this observation a step further, observing that users can often be re-identified using various forms of structural auxiliary information; these results were demonstrated using a real e-mail graph from Enron Corporation [123] and social network graphs from LiveJournal, Twitter, and Flickr [187]. We will return to an in-depth discussion of the state-of-the-art in privacy protection for social network graphs in Section 7.1. In addition to these examples, attacks on purportedly de-identified data sets have been illustrated in domains as diverse as GPS traces [120, 145] and genomic records [125, 170, 171, 172].

Note that not all attacks need to involve linking. Some involve reconstructing the original data to uncover pieces of information that are considered confidential. One such example was discussed by Meyer and Kadane [177] in relation to the 1990 decennial census. Two important uses of census data are distribution of federal funds and reapportionment (the assignment of seats in the House of Representatives to different states). Thus, undercounting different segments of the population (including minorities) is a serious political issue, and there is a debate about whether to adjust the census data to control for undercounting.

In 1991, the Commerce Department decided not to use the adjusted census data. It also refused to release the adjusted data. Following a congressional subpoena, a compromise was reached and the Commerce Department released adjusted population counts for every other census block and for all blocks whose adjusted population was at least 1,000 [177]. The leaders of the Florida House of Representatives asked Meyer and Kadane to reconstruct these missing values based on the actual census counts and on the released adjusted counts. Later, due to a lawsuit, the rest of the adjusted data was released and Meyer and Kadane were able to evaluate the accuracy of their reconstruction. Using relatively simple techniques based on comparisons of unadjusted counts for various blocks (see [177] for more details), they were able to obtain remarkably accurate results. For the 23 congressional districts of Florida that existed at the time, their estimate of the adjusted population differed from the official adjusted counts by at most 79 people. Meanwhile, the difference between the adjusted and unadjusted counts was on the order of several thousand people. Thus the Commerce Department's naive use of suppression ended up concealing less information than they intended.

Algranati and Kadane [19] discuss another example of data reconstruction. This time it involves the U.S. Department of Justice. In 2000, the U.S. Department of Justice released a report [248] about death penalty statistics for federal crimes. When a federal crime has been committed, the U.S. Attorney in charge of the case must make a recommendation on whether or not to seek the death penalty. The case is also reviewed by the Department of Justice, which also submits a recommendation. Finally, the Attorney General reviews the case and makes the final decision about this process (for more details about the circumstance of the report and the nature of the decisions, see [19, 248]). The Attorney General's decision is made public but the recommendations made by the U.S. Attorney and the Department of Justice are confidential. Algranati and Kadane focused on the 682 cases from 1995 to 2000 that are contained in this report. This report contains eight measured variables: the federal district, defendant's race, victim's race, the crime, whether or not there were multiple victims,

and the recommendations made by the U.S. Attorney, the Department of Justice, and the Attorney General. The data were released as a set of lower-dimensional tables of counts. Using some simple combinatorial techniques, Algranati and Kadane were able to fully recover 386 out of 682 records. They were also able to recover the combination of defendant race, federal district and all three recommendations for all of the 682 cases. Again, a naive release of data allowed for the recovery of most of the information that was considered confidential.

All of these examples serve to illustrate the challenges and importance of developing appropriate anonymization measures for published data.

## 1.3 Running Example

To prevent privacy breaches, organizations that want to publish data must resolve possible privacy issues before releasing data. We introduce privacy issues in data publishing by the following example scenario. A centralized *trusted* data collection agency, say Gotham City Hospital, collects information from a set of patients. The information collected from each patient consists of identifying information like name; demographic information like age, gender, zip code, and nationality; and the patient's medical condition. The data are put into a table like Table 1.1. Researchers in Gotham City University, who study how

Table 1.1. Medical record table.

| | Name | Age | Gender | Zip Code | Nationality | Condition |
|---|---|---|---|---|---|---|
| 1 | Ann | 28 | F | 13053 | Russian | Heart disease |
| 2 | Bruce | 29 | M | 13068 | Chinese | Heart disease |
| 3 | Cary | 21 | F | 13068 | Japanese | Viral infection |
| 4 | Dick | 23 | M | 13053 | American | Viral infection |
| 5 | Eshwar | 50 | M | 14853 | Indian | Cancer |
| 6 | Fox | 55 | M | 14750 | Japanese | Flu |
| 7 | Gary | 47 | M | 14562 | Chinese | Heart disease |
| 8 | Helen | 49 | F | 14821 | Korean | Flu |
| 9 | Igor | 31 | M | 13222 | American | Cancer |
| 10 | Jean | 37 | F | 13227 | American | Cancer |
| 11 | Ken | 36 | M | 13228 | American | Cancer |
| 12 | Lewis | 35 | M | 13221 | American | Cancer |

diseases correlate with patients' demographic attributes, can benefit substantially from analyzing these data and have made a request to the hospital for releasing the table. Now, the question is whether releasing Table 1.1 is safe. In fact, the hospital has a privacy policy that prevents it from releasing patients' identifying information. Obviously, releasing Table 1.1, which contains names, would violate this policy. However, does removal of names from Table 1.1 make the table safe for release? Consider a researcher, say Mark, who is a friend of Eshwar and knows that Eshwar is a 50-year-old Indian male having zip code 14853. He also knows that Eshwar visited Gotham City Hospital several times. If Mark saw this table with names removed, he would be almost sure that his friend Eshwar got cancer, because the fifth record is the only record that matches Mark's knowledge about Eshwar. Age, gender, zip code, and nationality are called *quasi-identifier attributes*, because by looking at these attributes an adversary may potentially identify an individual in the data set.

One way to prevent Mark from being able to infer Eshwar's medical condition is to make sure that, in the released data, no patient can be distinguished from a group of $k$ patients by using age, gender, zip code, and nationality. We call a table that satisfies this criterion a $k$-anonymous table. Table 1.2 is a modified version of the medical record table that is 4-anonymous, where names have been removed, age values have been generalized to age groups, gender values have been generalized to Any, zip codes have been generalized to first few digits and nationality values have been generalized to different geographical granularities. Now, when Mark sees this generalized table, he only knows that Eshwar's record is in the second group and is not sure whether Eshwar had flu or cancer. However, as will be seen later, this table is still not safe for release.

For now, let us assume that Gotham City Hospital somehow decides to consider 4-anonymous tables to be safe for release; but in addition to Table 1.2, there are many 4-anonymous tables which can be derived from the medical record table. Table 1.3 is another 4-anonymous table derived from the original medical record table. Which one should Gotham City Hospital choose to release? Intuitively, the hospital should choose the one that is the most useful for the researchers who request

Table 1.2. Generalized medical record table.

| | | Age | Gender | Zip Code | Nationality | Condition |
|---|---|---|---|---|---|---|
| (Ann) | 1 | 20–29 | Any | 130** | Any | Heart disease |
| (Bruce) | 2 | 20–29 | Any | 130** | Any | Heart disease |
| (Cary) | 3 | 20–29 | Any | 130** | Any | Viral infection |
| (Dick) | 4 | 20–29 | Any | 130** | Any | Viral Infection |
| (Eshwar) | 5 | 40–59 | Any | 14*** | Asian | Cancer |
| (Fox) | 6 | 40–59 | Any | 14*** | Asian | Flu |
| (Gary) | 7 | 40–59 | Any | 14*** | Asian | Heart disease |
| (Helen) | 8 | 40–59 | Any | 14*** | Asian | Flu |
| (Igor) | 9 | 30–39 | Any | 1322* | American | Cancer |
| (Jean) | 10 | 30–39 | Any | 1322* | American | Cancer |
| (Ken) | 11 | 30–39 | Any | 1322* | American | Cancer |
| (Lewis) | 12 | 30–39 | Any | 1322* | American | Cancer |

[a]No record can be distinguished from a group of four based on Age, Gender, Zip Code, and nationality.
[b]Names are removed. Age values are generalized to age groups. Gender values are generalized to Any. Zip codes are generalized to first few digits. Nationality values are generalized to different geographical granularities.

Table 1.3. Another generalized medical record table.

| | | Age | Gender | Zip Code | Nationality | Condition |
|---|---|---|---|---|---|---|
| (Ann) | 1 | 20–59 | F | 1**** | Any | Heart disease |
| (Helen) | 8 | 20–59 | F | 1**** | Any | Flu |
| (Cary) | 3 | 20–59 | F | 1**** | Any | Viral infection |
| (Jean) | 10 | 20–59 | F | 1**** | Any | Cancer |
| (Eshwar) | 5 | 20–59 | M | 1**** | Asian | Cancer |
| (Fox) | 6 | 20–59 | M | 1**** | Asian | Flu |
| (Gary) | 7 | 20–59 | M | 1**** | Asian | Heart disease |
| (Bruce) | 2 | 20–59 | M | 1**** | Asian | Heart Disease |
| (Igor) | 9 | 20–39 | M | 13*** | American | Cancer |
| (Dick) | 4 | 20–39 | M | 13*** | American | Viral infection |
| (Ken) | 11 | 20–39 | M | 13*** | American | Cancer |
| (Lewis) | 12 | 20–39 | M | 13*** | American | Cancer |

[a]The second record has been swapped with the eighth record, and the fourth record has been swapped with the tenth record.

for the data. Assume that the primary objective of the researchers is to understand how diseases correlated with genders. Thus, the researchers want as little replacement of a gender value by Any as possible. It should be easy to see that Table 1.3 is a better choice than Table 1.2 in terms of the number of replacements of gender values by Any.

## 1.4 Overview

Given a data set, privacy-preserving data publishing can be intuitively thought of as a game among four parties:

- **Data user**, like the researchers in Gotham City University, who wants to utilize the data.
- **Adversary**, like Mark in the running example, who wants to derive private information from the data.
- **Data publisher**, like Gotham City Hospital, who collects the data and wants to release the data in a way that satisfies the data user's need but also prevents the adversary from obtaining private information about the individuals in the data.
- **Individuals**, like Eshwar, whose data are collected by the data publisher. In some cases, the individuals agree with the data publisher's privacy policy, trust the data publisher and give the data publisher all the requested information. In these cases, it is the data publisher's responsibility to ensure privacy preservation. In other cases, the individuals do not trust the data publisher and want to make sure that the data publisher cannot precisely identify their sensitive information (e.g., by adding noise to their data records so that the data publisher can only have accurate aggregate statistics, but noisy individual data values). Although the primary focus of this paper is on trusted data publishers, we will also discuss untrusted data publishers in Section 4.2.

There is a fundamental tradeoff between privacy and utility. At one extreme, the data publisher may release nothing so that privacy is perfectly preserved; however, no one is able to use the data. At the other extreme, the data publisher may release the data set without any modification so that data utility can be maximized; however, no privacy protection is provided. For the data publisher to release useful data in a way that preserves privacy, the following three components need to be defined.

- **Sanitization mechanism:** Given an *original data set*, e.g., Table 1.1, a sanitization mechanism sanitizes the data set by making the data less precise. This mechanism defines the space of possible "snapshots" of the original data set that are considered as candidates for release. We call such a snapshot a *release candidate*. Generalization is an example sanitization mechanism. Tables 1.2 and 1.3 are two release candidates of such a mechanism when applied to Table 1.1. We will first introduce some common sanitization mechanisms in Section 1.5 and have an in-depth discussion in Section 4.
- **Privacy criterion:** Given a release candidate, the privacy criterion defines whether the release candidate is safe for release or not. $k$-Anonymity is an example privacy criterion. Privacy criteria are the focus of Section 2.
- **Utility metric:** Given a release candidate, the utility metric quantifies the utility of the release candidate (equivalently, the information loss due to the sanitization process). For example, the researchers in Gotham City University use the number of replacements of gender values by Any as their utility measure. We survey utility metrics in Section 3.

Given the above three components, one approach to privacy-preserving data publishing is to publish the most useful release candidate that satisfies the privacy criterion. An algorithm that takes an original data set and generates a release candidate that satisfies a given privacy criterion while providing high utility[2] is called an *anonymization (or sanitization) algorithm*. The terms "anonymization" and "sanitization" will be used interchangeably. A selected list of interesting anonymization algorithms is presented in Section 4.

After the data publisher finds a good release candidate and makes it public, the data user will use it for good and the adversary will attack it. Because the sanitization mechanism has perturbed the data to make it less precise and less sensitive, the data user may not be able

[2] Note that providing the maximum utility among all release candidates may not be algorithmically feasible and may also be undesirable because it gives an adversary an additional avenue of attack (see Section 6).

to use the data in a straightforward manner. For example, suppose that Table 1.3 is released, and the data user wants to know the fraction of patients with ages between 20 and 30 who have heart disease. This query cannot be answered precisely based on Table 1.3, but may be answered probabilistically. A methodology is needed to answer such queries in a meaningful and consistent manner. In addition to database queries, the data user may also want to build machine-learning models (for a prediction task) or conduct statistical analysis (to test whether a finding from a sanitized data set is statistically significant). We will discuss how to do so in Section 5 and point the readers to related literature.

From the adversary's point of view, although the released data satisfy a privacy criterion (or a few criteria), it is still possible to uncover some individuals' sensitive information. This is because each privacy criterion has its own assumption and sometimes only protects data against a few types of attacks. For example, Table 1.2 satisfies the $k$-anonymity criterion. However, it is vulnerable to a *homogeneity attack*: although no one cannot distinguish Jean's record from the other three records (Igor's, Ken's, and Lewis') based on the quasi-identifier attributes, we are 100% sure that she has cancer (if we know her quasi-identifier attributes and the fact that her data are in Table 1.2). Furthermore, some anonymization algorithm have special behavior that may allow the adversary to make further inference about the data, and the adversary may have more background knowledge than a privacy criterion assumes. We review interesting attacks against sanitized data in Section 6.

We note that there can potentially be multiple data users with different data needs, multiple adversaries with different purposes and knowledge about individuals in the data, and multiple data publishers (whose data sets may overlap with each other) who would like to release versions of their data. A single data publisher may also want to release different versions of the data at different times. Furthermore, the original data set may not be a single table; it may be a relational database (that contains multiple tables), a market-basket database (in which each record is a set of items), a search log (in which each record is a search query with some metadata), a social network

(relating individuals), and so on. These variations all add to the complexity of the problem and will be addressed with different levels of details (in proportion to the progress that has been made on these problems). In particular, we discuss social network privacy in Section 7.1, search log privacy in Section 7.2, location privacy of mobile applications in Section 7.3, and challenges for future research in Section 7.4.

## 1.5 Examples of Sanitization Mechanisms

Before proceeding to the next chapter, we will first briefly introduce a number of common sanitization mechanisms to facilitate our discussion. It is important to have a basic idea of such mechanisms because a privacy criterion is defined on the output of such a mechanism, an adversary breaches privacy by analyzing such an output, and a data user studies such an output. However, we do not try to cover all of the sanitization mechanisms here. An in-depth discussion of mechanisms and algorithms will be presented in Section 4.

Recall that a sanitization mechanism defines the space of all possible release candidates in an application of privacy-preserving data publishing. An anonymization algorithm finds a release candidate that is both useful and safe (according to a given privacy criterion) from this space. To simplify our discussion, we consider the original data set to be a table (e.g., Table 1.1), in which each column is an attribute and each row is the data record of an individual. Other kinds of data (sets of items, text data, graph and network data, and others) will be discussed later (primarily in Section 7).

**Generalization:** The generalization mechanism produces a release candidate by generalizing (coarsening) some attribute values in the original table. We have seen two examples of such release candidates in Tables 1.2 and 1.3. The basic idea is that, after generalizing some attribute values, some records (e.g., Ann's record and Bruce's record in Table 1.2) would become identical when projected on the set of quasi-identifier (QI) attributes (e.g., age, gender, zip code, and nationality). Each group of records that have identical QI attribute values is called an *equivalence class*.

**Suppression:** The suppression mechanism produces a release candidate by replacing some attribute values (or parts of attribute values) by a special symbol that indicates that the value has been suppressed (e.g., "*" or "Any"). Suppression can be thought of as a special kind of generalization. For example, in Table 1.2, we can say that some digits of zip codes and all the gender values have been suppressed.

**Swapping:** The swapping mechanism produces a release candidate by swapping some attribute values. For example, consider Table 1.1. After removing the names, the data publisher may swap the age values of Ann and Eshwar, swap the gender values of Bruce and Cary, and so on.

**Bucketization:** The bucketization mechanism produces a release candidate by first partitioning the original data table into non-overlapping groups (or buckets) and then, for each group, releasing its projection on the non-sensitive attributes and also its projection on the sensitive attribute. Table 1.4 is a release candidate of the bucketization mechanism when applied to Table 1.1. In this case the Condition attribute is considered to be sensitive and the other attributes are not. The idea is that after bucketization, the sensitive attribute value of an individual would be indistinguishable from that of any other individual in the same group. Each group is also called an *equivalence class*.

Table 1.4. Bucketized medical record table.

| | Age | Gender | Zip Code | Nationality | BID | BID | Condition |
|---|---|---|---|---|---|---|---|
| (Ann) | 28 | F | 13053 | Russian | 1 | 1 | Heart disease |
| (Bruce) | 29 | M | 13068 | Chinese | 1 | 1 | Heart disease |
| (Cary) | 21 | F | 13068 | Japanese | 1 | 1 | Viral infection |
| (Dick) | 23 | M | 13053 | American | 1 | 1 | Viral infection |
| (Eshwar) | 50 | M | 14853 | Indian | 2 | 2 | Cancer |
| (Fox) | 55 | M | 14750 | Japanese | 2 | 2 | Flu |
| (Gary) | 47 | M | 14562 | Chinese | 2 | 2 | Heart disease |
| (Helen) | 49 | F | 14821 | Korean | 2 | 2 | Flu |
| (Igor) | 31 | M | 13222 | American | 3 | 3 | Cancer |
| (Jean) | 37 | F | 13227 | American | 3 | 3 | Cancer |
| (Ken) | 36 | M | 13228 | American | 3 | 3 | Cancer |
| (Lewis) | 35 | M | 13221 | American | 3 | 3 | Cancer |

[a]Three buckets are created and identified by their bucket IDs (BID).
[b]A patient's condition in a bucket is indistinguishable from any other patient's condition in the same bucket.

Table 1.5. Randomized medical record table.

| | | Age | Gender | Zip code | Nationality | Condition |
|---|---|---|---|---|---|---|
| (Ann) | 1 | 30 | F | 13073 | Russian | Heart disease |
| (Bruce) | 2 | 28 | M | 13121 | American | Heart disease |
| (Cary) | 3 | 22 | M | 13024 | Japanese | Cancer |
| (Dick) | 4 | 20 | M | 13030 | American | Viral infection |
| ... | | ... | ... | ... | ... | ... |

[a]Names are removed. Random noise is added to each attribute value. For numeric attributes (age and zip code), Gaussian noise is added. For categorical attributes (gender, zip code, and nationality), with some probability, an attribute value is replaced by a random value in the domain.

**Randomization:** A release candidate of the randomization mechanism is generated by adding random noise to the data. The sanitized data could be sampled from a probability distribution (in which case it is known as *synthetic data*) or the sanitized data could be created by randomly perturbing the attribute values. For example, Table 1.5 is such a release candidate for Table 1.1, where random noise is added to each attribute value. We add Gaussian noise with mean 0 and variance 4 to age and also Gaussian noise with 0 mean and variance 500 to zip code. For gender, nationality, and condition, with probability 1/4, we replace the original attribute value with a random value in the domain; otherwise, we keep the original attribute value. Note that, in general, we may add different amounts of noise to different records and different attributes. Several application scenarios of randomization can be distinguished. In *input randomization*, the data publisher adds random noise to the original data set and releases the resulting randomized data, like Table 1.5. In *output randomization*, data users submit queries to the data publisher and the publisher releases randomized query results. In *local randomization*, individuals (who contribute their data to the data publisher) randomize their own data before giving their data to the publisher. In this last scenario, the data publisher is no longer required to be trusted.

**Multi-view release:** To increase data utility, the data publisher may release multiple views of a single original data set, where the released views are outputs of one (or more) of the above sanitization mechanisms. For example, a release candidate could be a set of generalized tables. As a special case of multiple generalized tables, we show an

Table 1.6. An example of multi-marginal release.

(a) Marginal on gender, nationality

| Gender | Nationality | Count |
|---|---|---|
| F | Russian | 1 |
| F | Japanese | 1 |
| F | Korean | 1 |
| F | American | 1 |
| M | Chinese | 2 |
| M | American | 4 |
| M | Indian | 1 |
| M | Japanese | 1 |

(b) Marginal on gender, condition

| Gender | Condition | Count |
|---|---|---|
| F | Heart disease | 1 |
| F | Viral infection | 1 |
| F | Flu | 1 |
| F | Cancer | 1 |
| M | Heart disease | 2 |
| M | Viral infection | 1 |
| M | Flu | 1 |
| M | Cancer | 4 |

example of *multi-marginal release* in Table 1.6, which consists of two views of the original data Table 1.1. Each view is generated by projecting the original data table on a subset of attributes and computing the counts. Such a view is called a marginal table or a histogram on the subset of attributes.

# 2

# Privacy Definitions

> All the evolution we know of proceeds from the vague to the definite.
>
> — Charles Sanders Peirce

Intuition and conventional wisdom have long indicated that the privacy of individuals can be protected by coarsening personal data, adding random noise (to the data themselves, or to the output of aggregate queries), swapping attribute values amongst individuals' records, or removing small counts in published contingency tables. A large body of recent work has begun to formalize this intuition, providing numerous definitions of privacy, and characterizing the nature of the information protected. In this section, we will describe formal privacy definitions that, we believe, represent milestones in the literature. These definitions include Samarati and Sweeney's $k$-anonymity [226, 241], Machanavajjhala et al.'s $\ell$-diversity [166], Martin et al.'s $(c, k)$-safety [173] and Chen et al.'s 3D privacy criterion [51] (against Boolean background knowledge), Evfimievski et al.'s $(\alpha, \beta)$-privacy and $\gamma$-amplification [92] (against probabilistic background knowledge), Dwork's differential privacy [85], and Shannon's perfect secrecy [232] (equivalent to Miklau's

perfect privacy [179]). Beyond these milestones, a selected list of other privacy definitions will be briefly summarized in Section 2.8. Then, we conclude this section with a unified framework whose aim is to allow one to compare different privacy definitions on the same basis.

## 2.1 Disclosure Risk

Before describing what we consider to be milestones in the recent development of formal privacy definitions, we first discuss work on measuring *disclosure risk*. Disclosure risk is a term frequently used in the official statistics literature to refer to quantifiable estimates of the possibility of a privacy breach. It has long been known that publishing data collected from individuals can potentially breach privacy even when identifier attributes are removed from the data. Techniques like coarsening personal data, adding random noise, swapping attribute values, removing small counts in published contingency tables, and generating synthetic data were proposed to address this problem. However, to ensure privacy preservation, data publishers must at least be able to measure the disclosure risk of the outputs from these techniques. Measuring disclosure risk is a key step in defining privacy criteria. Many privacy criteria are defined based on placing a threshold on a measure of disclosure risk. It is, of course, important to distinguish measures of disclosure risk and privacy definitions from the mechanisms used to sanitize data since the quantity we measure (amount of privacy) should not depend on how we choose to represent the data.

There is a large body of work on measuring and estimating disclosure risk in data publishing. Since our main interest is in the resulting privacy definitions, we will only briefly discuss a small number of studies to set up the background for the privacy definitions to be introduced later in this section. A survey of disclosure risk measures is beyond the scope of this paper, and the interested reader can consult [264, 263].

**Small counts in contingency tables:** Consider releasing a contingency table; for each combination of attribute values in the table, we release the number of individuals having that combination. One could also release marginals of the table (i.e., the counts associated with various subsets of the attributes). One measure of disclosure risk is the

smallest count in the table or in a marginal. A small count indicates a rare combination of attributes and may lead to the re-identification of the associated individuals. Fellegi [97] discussed this kind of disclosure risk in the early 1970s and pointed out that, in the case of only releasing marginals of the contingency table, even if all the marginal counts are large, small counts in the contingency table may still be reconstructed by solving a system of linear equations. Recent work on bounding counts in contingency tables for marginal releases includes [75, 77, 78, 235]. Also, models have been proposed for estimating population counts (number of individuals in the population having a combination of attribute values) from the counts in a contingency table [30, 103]. This allows one to distinguish between population uniques and sample uniques.

**Identification rules for microdata:** Another way of measuring disclosure risk is by designing rules that can be used to identify individuals in a sanitized data set and assessing the effectiveness of the rules (for example, by counting how many individuals can be identified by the rules with high confidence). Spruill [236], in the early 1980s, suggested a distance-based method. For each sanitized record, compute the Euclidean distance between the sanitized record and each of the records in the original data set. Spruill then defines the disclosure risk as the sampling fraction (fraction of the original records released) times the percentage of sanitized records whose nearest neighbors in the original data set are their own original records.

Since then, there have been many studies based on identification rules. For example, Lambert [148] discussed some definitions of disclosure risk using rules based on probabilistic models. Let $r$, $i$, $j$, and $N$ denote a rule, a record in the sanitized data set, an individual in the population, and the population size, respectively. She defined the *worst-case risk* as

$$\max_j \max_i \Pr(\text{record } i \text{ is individual } j\text{'s record by rule } r),$$

the *average risk* as

$$(1/N) \sum_{j=1}^{N} \max_i \Pr(\text{record } i \text{ is individual } j\text{'s record by rule } r),$$

and a threshold-based risk as the fraction of individuals $j$ in the population with

$$\max_i \Pr(\text{record } i \text{ is individual } j\text{'s record by rule } r) \geq \tau,$$

for a given threshold $\tau$. For an overview of different identification rules, see [268, 269, 270].

**Decision-theoretic approach:** To give disclosure risk a theoretic foundation, Duncan and Lambert [83] in the late 1980s proposed the use of the decision theory. Suppose the adversary's "target" is $t$ (the target could be the identity of an individual, an attribute value, or a property of the original data set). Let $x$ denote a possible value of the target and $p_t(x \mid D^*)$ denote the probability density that the target $t$ has value $x$ after observing the sanitized data set $D^*$. In a very general sense, one can define a loss function $L_t(\delta, x)$ that represents the adversary's loss if he/she decides the target $t$ has value $\delta$, and the true value is $x$. Since the adversary does not know the true value, he/she would take the decision $\delta$ that minimizes the expected loss; this minimum expected loss is the adversary's "uncertainty" $U_t$ about $t$ after seeing $D^*$ and is written as:

$$U_t(D^*) = \min_\delta \int L_t(\delta, x) p_t(x|D^*) dx.$$

Observe that when $L_t(\delta, x) = (\delta - x)^2$ (for a numerical target $t$), the optimal decision is $\delta = E[x|D^*]$ and $U_t(D^*)$ is the variance of $t$ (given $D^*$). When $L_t(\delta, x) = -\log p_t(x|D^*)$ (for a categorical target $t$), $U_t(D^*)$ is the entropy of $t$ (given $D^*$). Variance and entropy are two common uncertainty functions which measure the disclosure risk for $t$ caused by releasing $D^*$. Small uncertainty represents high risk. For microdata, Duncan and Lambert [83] provided two other appropriate loss functions (hence two risk measures), which we omit. It is easy to see that when there are multiple possible targets, the worst-case disclosure risk can be measured by the inverse of $\min_t U_t(D^*)$. Interestingly, many modern privacy criteria (that will be described later) can be viewed as instantiations of this framework, requiring $\min_t U_t(D^*) \geq \theta$, for some threshold $\theta$, with different loss functions, probabilistic models and, in some cases, with background knowledge.

## 2.2 $k$-Anonymity

Sweeney in [241] demonstrated that releasing a data table by simply removing identifiers (e.g., names and social security numbers) can seriously breach the privacy of individuals whose data are in the table. By combining a public voter registration list and a released medical database of health insurance information, she was able to identify the medical record of the governor of Massachusetts. In fact, according to her study of the 1990 census data [240], 87% of the population of the United States can be uniquely identified on the basis of their five-digit zip code, gender, and date of birth.

This kind of attack is called *linking attack* (see Section 1.2). Take Table 1.1 for example. Suppose that we remove the Name attribute and release the resulting table. It is common that the adversary has access to several public databases. For instance, he can easily obtain a public voter registration list as shown in Table 2.1. Assume the area of zip code 13068 is a small town and Ann is the only 28-year-old female living in that town. When the adversary looks at Table 1.1 with names removed, he can almost be sure that the first record with *Age* = 28, *Gender* = F, and *Zip code* = 13068 is Ann's record by matching that record with Ann's record in the voter registration list. The goal of a linking attack is to find the identity of an individual in a released data set that contains no identifying attributes by linking the records in the data set to a public data set that contains identifying attributes. This linkage is performed with a set of *quasi-identifier* (QI) attributes that are in both data sets. In the above example, *Age*, *Gender*, and *Zip code* are QI attributes.

To protect data from linking attacks, Samarati and Sweeney proposed $k$-anonymity [226, 241]. Let $D$ (e.g., Table 1.1) denote the

Table 2.1. Example voter registration list.

| Name | Age | Gender | Zip code |
|---|---|---|---|
| Ann | 28 | F | 13068 |
| Bob | 21 | M | 13068 |
| Carol | 24 | F | 13068 |
| Dan | 21 | M | 13068 |
| Ed | 52 | M | 13068 |
| ... | ... | ... | ... |

original data table and $D^*$ (e.g., Table 1.2) denote a release candidate of $D$ produced by the generalization mechanism.

---

**Definition 2.1 ($k$-Anonymity).** Given a set of QI attributes $Q_1, \ldots, Q_d$, release candidate $D^*$ is said to be $k$-anonymous with respect to $Q_1, \ldots, Q_d$ if each unique tuple in the projection of $D^*$ on $Q_1, \ldots, Q_d$ occurs at least $k$ times.

---

Table 1.2 is 4-anonymous. Now, no matter what public databases the adversary has access to, he can only be sure that Ann's record is one of the first four. While $k$-anonymity successfully protects data from linking attacks, an individual's private information can still leak out. For example, the last four individuals of Table 1.2 have cancer. Although the adversary is not able to know which record belongs to Jean, he is sure that Jean has cancer if he knows Jean's age, gender, and zip code from a public database. This motivated Machanavajjhala et al., who propose the principle of $\ell$-diversity, which is presented in the next section.

In practice, multiple criteria should be enforced at the same time in order to protect data from different kinds of attacks. We note that, for a given data publication scenario, the issues of setting the parameter $k$ and deciding which attributes to include in the set of QI attributes have not been well-addressed in the literature. For the second question, a simple approach that has often been taken is to conservatively include all of the non-sensitive attributes in the set of QI attributes. However, further research is still needed to develop principles to help determine the right $k$ value for a given scenario; we briefly return to this question in Section 7.4.3.

## 2.3 $\ell$-Diversity

$k$-Anonymity ensures that individuals cannot be uniquely re-identified in a data set and thus guards against linking attacks. However, Machanavajjhala et al. [166, 168] showed that adversaries with more background knowledge, also called *adversarial knowledge*, can infer sensitive information about individuals even without re-identifying them.

The following two attacks — *homogeneity attack* and *background knowledge attack* — presented in that paper illustrate such adversaries.

We already encountered the homogeneity attack in the previous section. Recall the case of Jean from Table 1.2. Her neighbor Alice knows that Jean is a 37-year-old American woman from zip code 13227. If Alice knows that Jean is in the table, then she knows that Jean's information resides in one of the last four tuples in the table. Though Alice cannot uniquely identify Jean's record, she knows that Jean has cancer thus breaching Jean's privacy.

Next suppose Alice knows that her pen friend Cary, who is a 21-year-old Japanese living in zip code 13068, is also admitted to the hospital. Unlike in the previous case, given only this information Alice can only deduce that Cary has either the heart disease or the viral infection. However, it is well-known in medical circles that 25-year-old Japanese have a very low incidence of heart disease due to their diet. Thus Alice can deduce that Cary is much more likely to have the viral infection rather than the heart disease and breach her privacy. Machanavajjhala et al. identified the importance of incorporating adversarial background knowledge into a privacy metric and proposed the Bayes-Optimal privacy and the principle of $\ell$-diversity. More complex forms of background knowledge attacks will be described in the later sections.

In order to guarantee privacy against such adversaries, Machanavajjhala et al. first propose a formal but impractical definition of privacy called Bayes-Optimal privacy. The attributes in the input table are considered to be partitioned into non-sensitive QI attributes (called $Q$) and sensitive attributes (called $S$). The adversary is assumed to know the complete joint distribution $f$ of $Q$ and $S$. Publishing a generalized table breaches privacy according to Bayes-Optimal privacy if the adversary's prior belief in an individual's sensitive attribute is very different from the adversary's posterior belief after seeing the published generalized table. More formally, adversary Alice's prior belief, $\alpha_{(q,s)}$, that Bob's sensitive attribute is $s$ given that his non-sensitive attribute is $q$, is her background knowledge:

$$\alpha_{(q,s)} = P_f(t[S] = s \mid t[Q] = q) = \frac{f(s,q)}{\sum_{s' \in S} f(s,q)},$$

where $t[S]$ and $t[Q]$ denote the sensitive value and the vector of QI attribute values of individual $t$, respectively; $P_f$ denotes the probability computed based on distribution $f$. On observing the published table $T^\star$ which is generalized from $T$, and in which Bob's quasi-identifier $q$ has been generalized to $q^*$, her posterior belief about Bob's sensitive attribute is denoted by $\beta_{(q,s,T^\star)}$ and is equal to:

$$\beta_{(q,s,T^\star)} = P_f(t[S] = s \mid t[Q] = q \text{ and } T^\star \text{ and } t \in T)$$

Given the joint distribution $f$ and the output table $T^\star$, Machanavajjhala et al. derived a formula for $\beta_{(q,s,T^\star)}$.

---

**Theorem 2.1 (from [166]).** Let $T^\star$ be a published table which is obtained by performing generalizations on a table $T$; let $X$ be an individual with $X[Q] = q$ who appears in the table $T$ (and also $T^\star$); let $q^\star$ be the generalized value of $q$ in $T^\star$; let $s$ be a possible value of the sensitive attribute; let $n_{(q^\star,s')}$ be the number of tuples $t^\star \in T^\star$ where $t^\star[Q] = q^\star$ and $t^\star[S] = s'$; and let $f(s' \mid q^\star)$ be the conditional probability of the sensitive attribute being $s'$ conditioned on the fact that the non-sensitive attribute $Q$ is some $q'$ which can be generalized to $q^\star$. Then the posterior belief that $X[S] = s$ after observing $T^\star$ is given by:

$$\beta_{(q,s,T^\star)} = \frac{n_{(q^\star,s)} \frac{f(s|q)}{f(s|q^\star)}}{\sum_{s' \in S} n_{(q^\star,s')} \frac{f(s'|q)}{f(s'|q^\star)}} \tag{2.1}$$

---

Publishing a table $T^\star$ satisfies Bayes-Optimal privacy if the distance between $\alpha_{(q,s)}$ and $\beta_{(q,s,T^\star)}$ is small for every $q \in Q$ and for every $s \in S$; where distance is measured either using the difference or ratio of the two quantities.

However, Bayes-Optimal privacy has the following limitations. First, the data publisher is unlikely to know the full distribution $f$. Second, it is unlikely that the adversary knows the entire joint distribution either. Further, the data publisher may not know the exact knowledge the adversary possesses. For instance, Alice knew that Cary had a very low incidence of heart disease; but the data publisher may not know this. Third, the above analysis captures only distributional

knowledge and does not capture instance level knowledge. For instance, Alice may know that Igor has heart disease by talking to his wife. Next, there will be multiple adversaries, each with varying amounts of knowledge about the individuals in the table and the joint distribution; the data publisher would have to be able to specify which of these adversaries are guarded against. Finally, checking the Bayes-Optimal condition for every $(q, s)$ combination in the domain might be computationally tedious.

To overcome the limitations of Bayes-Optimal privacy, Machanavajjhala et al. proposed the $\ell$-diversity principle, which is motivated by the fact that Bayes-Optimal privacy is not satisfied (a) when there is lack of diversity in the sensitive values within a group of tuples sharing the same QI values (like in the homogeneity attack), and (b) when the adversary is able to eliminate all but one of the sensitive values associated with the group (like in the background knowledge attack). A table is said to satisfy the $\ell$-diversity principle if every group of tuples that share the same QI values in the table have at least $\ell$ *well-represented* sensitive values; i.e., there are at least $\ell$-distinct sensitive values that are of roughly equal proportion. Table 2.2 is an example of a 3-diverse table.

This principle and the associated notion of *well-representedness* can be instantiated in many ways. One instantiation is called *entropy*

Table 2.2. A 3-diverse generalized table.

| | | Age | Gender | Zip code | Nationality | Condition |
|---|---|---|---|---|---|---|
| (Ann) | 1 | 20–59 | F | 1**** | Any | Heart disease |
| (Helen) | 8 | 20–59 | F | 1**** | Any | Flu |
| (Cary) | 3 | 20–59 | F | 1**** | Any | Viral infection |
| (Jean) | 10 | 20–59 | F | 1**** | Any | Cancer |
| (Eshwar) | 5 | 20–59 | M | 1**** | Asian | Cancer |
| (Fox) | 6 | 20–59 | M | 1**** | Asian | Flu |
| (Gary) | 7 | 20–59 | M | 1**** | Asian | Heart disease |
| (Ken) | 11 | 20–39 | M | 13*** | American | Cancer |
| (Igor) | 9 | 20–39 | M | 13*** | American | Cancer |
| (Dick) | 4 | 20–39 | M | 13*** | American | Viral infection |
| (Bruce) | 2 | 20–59 | M | 1**** | Asian | Heart disease |
| (Lewis) | 12 | 20–39 | M | 13*** | American | Cancer |

[a]Each 4 anonymous group of tuples has at least three distinct sensitive values of roughly equal proportions.
[b]The above table is 1.5-entropy diverse, and is recursive (2,3)-diverse.

*$\ell$-diversity*, where in each group of tuples with the same QI value, the entropy of the sensitive attribute should be at least $\log \ell$. Entropy $\ell$-diversity was first proposed by Ohrn and Ohno-Machado [194] as a way of defending against the homogeneity problem (without considering the role of background knowledge).

Another instantiation of the $\ell$-diversity principle is captured by *recursive* $(c, \ell)$ *diversity*. Let $s_1, \ldots, s_m$ be the possible values of the sensitive attribute $S$ in a group of tuples with generalized QI value $q^\star$, henceforth called a $q^\star$-block (which is also called a equivalence class). Assume that we sort the counts $n_{(q^\star, s_1)}, \ldots, n_{(q^\star, s_m)}$ in descending order and name the elements of the resulting sequence $r_1, \ldots, r_m$. $\ell$-Diversity can also be interpreted as follows: an adversary can breach the privacy of a $\ell$-diverse $q^\star$-block only if he/she can eliminate at least $\ell - 1$ possible values of $S$. That is, in a 2-diverse table, none of the sensitive values should appear too frequently. A $q^\star$-block is defined to be $(c, 2)$-diverse if $r_1 < c(r_2 + \cdots + r_m)$ for some user-specified constant $c$. For $\ell > 2$, we say that a $q^\star$-block satisfies *recursive* $(c, \ell)$*-diversity* if we can eliminate one possible sensitive value in the $q^\star$-block and still have a $(c, \ell - 1)$-diverse block. This recursive definition can be succinctly stated as follows:

---

**Definition 2.2 (Recursive $(c, \ell)$-Diversity).** In a given $q^\star$-block, let $r_i$ denote the number of times the $i$-th most frequent sensitive value appears in that $q^\star$-block. Given a constant $c$, the $q^\star$-block satisfies *recursive* $(c, \ell)$*-diversity* if $r_1 < c(r_\ell + r_{\ell+1} + \cdots + r_m)$. A table $T^\star$ satisfies recursive $(c, \ell)$-diversity if every $q^\star$-block satisfies recursive $\ell$-diversity. We say that 1-diversity is always satisfied.

---

The recursive $(c, \ell)$-diversity, thus, can be interpreted in terms of adversarial background knowledge. It guards against all adversaries who possess at most $\ell - 2$ statements of the form "Bob does not have heart disease". We call such statements *negation* statements.

At this point we would like to remind the readers the question we raised in the previous section: "How does a data publisher decide which attributes should be included in the set of QI attributes?" QI attributes are just a special case of background knowledge. $k$-Anonymity only

considered background knowledge quantified by the set of QI attributes. $\ell$-Diversity considered adversaries possessing negation statements in addition to QI attributes. In the next few sections, we will describe formal models of adversarial background knowledge and progressively more complex forms of background knowledge.

## 2.4 Protection Against Boolean Background Knowledge

$\ell$-Diversity highlighted the importance of a formal specification of the background knowledge available to the adversary, and the fact that the data publisher may not know the adversarial background knowledge but may still be able to guard against it. Motivated by this, Martin et al. [173] and Chen et al. [51] considered more general forms of background knowledge. In general, one can describe background knowledge using Boolean logic sentences and seek to provide privacy protection against an adversary who knows a certain number of such sentences. Martin et al. first introduced a privacy framework based on such an idea. Then, Chen et al. provided a privacy criterion that is easily understandable (i.e., can be explained precisely using plain English) and encompasses $k$-anonymity and $(c, \ell)$-diversity as special cases.

Consider the running example. Let $D$ denote the original medical record table (Table 1.1). After applying a sanitization procedure, the data publisher obtains a generalized view of $D$ (Table 1.2), denoted by $D^*$. To understand whether $D^*$ is safe for release, we consider an adversary whose goal is to predict (or infer) whether a target individual $t$ (say, Eshwar) has a target sensitive value $s$ (say, cancer). In making this prediction, the adversary would be assumed to have access to the release candidate $D^*$, as well as his own knowledge $K$. This knowledge may include information from similar data sets released by other organizations, social networks relating individuals, and other instance-level information. A robust privacy criterion should place an upper bound on the adversary's confidence in predicting any individual $t$ to have sensitive value $s$. In other words, the criterion should guarantee that, for any $t$ and $s$, $\Pr(t \text{ has } s \,|\, K, D^*) < c$, for some threshold value $c$. It is equivalent to say

$$\max_{t,s} \Pr(t \text{ has } s \,|\, K, D^*) < c.$$

We call $\max_{t,s} \Pr(t \text{ has } s \,|\, K, D^*)$ the *breach probability*, which represents the adversary's confidence in predicting the sensitive value $s$ of the least protected individual $t$ when the adversary has knowledge $K$ and obtains release candidate $D^*$.

Returning to the example, assume that each individual has only one disease in $D$. In the absence of adversarial knowledge, intuitively the adversary can predict Eshwar to have cancer with confidence $\Pr(\text{Eshwar has Cancer } \,|\, D^*) = 1/4$ because there are four individuals in Eshwar's equivalence class, only one of whom has cancer; without additional knowledge, no one is more likely than the other. However, the adversary can improve his confidence if he has some additional knowledge. For example:

- The adversary knows Eshwar personally, and is sure that he does not have heart disease. After removing the record with heart disease, the probability that Eshwar has cancer becomes 1/3.
- From another data set, the adversary determines that Fox has Flu. By further removing Fox's Flu record, the probability that Eshwar has cancer becomes 1/2.

In defining a privacy criterion incorporating such background knowledge, two key problems need to be addressed. First, one must provide the data publisher with the means to specify adversarial knowledge $K$. Second, one must compute the breach probability (in a computationally efficient way).

### 2.4.1 Specification of Adversarial Knowledge

We will first discuss how the adversarial knowledge can be specified, and then we will discuss how to compute breach probabilities. We note that computation of breach probabilities under general Boolean logic sentences is NP-hard [173]. That means our focus should be on special logic sentences that are efficiently computable and represent useful adversarial knowledge.

The problem of adversarial-knowledge specification is further complicated by the fact that, in general, the data publisher does not know

precisely what knowledge an adversary has. To address this, Martin et al. proposed the use of a *language* for expressing such knowledge. Because it is nearly impossible for the data publisher to anticipate specific adversarial knowledge, they instead propose to quantify the *amount* of knowledge an adversary could have, and to release data that are resilient to a certain amount of knowledge regardless of the specific content of this knowledge. Specifically, they define the language $\mathcal{L}_{basic}(k)$ to be the set of all possible conjunctions of $k$ implications (i.e., $k$ implications connected by "and"). Each implication is of the following form:

$$[(u_1 \text{ has } v_1) \text{ and } \ldots \text{ and } (u_m \text{ has } v_m)]$$
$$\text{implies } [(t_1 \text{ has } s_1) \text{ and } \ldots \text{ and } (t_n \text{ has } s_n)],$$

where $u_i$ and $t_j$ are individuals in $D$, $v_i$ and $s_j$ are sensitive values, and $m$ and $n$ can be any positive numbers. An example logic sentence in $\mathcal{L}_{basic}(2)$ is

[((Fox has Flu) and (Igor has Cancer)) implies (Ken has Cancer)]
and
[(Helen has Flu) implies ((Fox has Flu) and (Lewis has Cancer))]

---

**Definition 2.3 ($(c,k)$-Safety).** Given knowledge threshold $k > 0$ and confidence threshold $c \in [0,1]$, release candidate $D^*$ is $(c,k)$-safe if

$$\max_{t \in T, s \in S, K \in \mathcal{L}_{basic}(k)} \Pr(t \text{ has } s \,|\, K, D^*) < c,$$

where $T$ is the set of individuals involved in $D$ and $S$ is the set of sensitive attribute values.

---

Chen et al. argued that $\mathcal{L}_{basic}(k)$ is not intuitive. It is difficult for the data publisher to understand the practical meaning of a conjunction of $k$ implications, thus making it hard to set an appropriate $k$ value in practice. Instead, they proposed to quantify possible adversarial knowledge from three intuitive dimensions. Suppose that the adversary's target is to determine whether individual $t$ has sensitive value $s$. They define the language $\mathcal{L}_{t,s}(\ell, k, m)$ to be the set of all logic sentences,

each of which represents an adversary that knows: (1) $\ell$ sensitive values that the target individual $t$ does not have, (2) the sensitive values of $k$ other individuals, and (3) $m$ individuals in $t$'s *same-value family* for a sensitive value $s$ (meaning that we can be sure that $t$ has sensitive value $s$ if any one of those $m$ individuals has $s$, especially if $s$ is a contagious disease).

---

**Definition 2.4 (Basic 3D privacy criterion).** Given knowledge threshold $(\ell, k, m)$ and confidence threshold $c \in [0, 1]$, release candidate $D^*$ is safe if

$$\max_{t \in T, s \in S, K \in \mathcal{L}_{t,s}(\ell,k,m)} \Pr(t \text{ has } s \,|\, K, D^*) < c,$$

where $T$ is the set of individuals involved in $D$ and $S$ is the set of sensitive attribute values.

---

Note that, for simplicity, we slightly modified the definition of the basic 3D privacy criterion of Chen et al. In the original definition, one can have possibly different $(\ell, k, m)$ and $c$ values for different sensitive values in order to give some sensitive values (e.g., AIDS as opposed to Flu) more protection. We also note that Chen et al. extended the basic 3D privacy criterion to a skyline privacy criterion, which provides the data publisher further flexibility, and studied set-valued sensitive attributes and different kinds of schema-level constraints.

To make the Boolean background knowledge used in $k$-anonymity and $\ell$-diversity explicit, Chen et al. showed that $k$-anonymity is a special case of the basic 3D privacy criterion where the identities of the individuals in the data set are considered to be the sensitive values, the knowledge threshold is $(0,\ k-2,\ 0)$ and the confidence threshold is 1, for all sensitive values. They also showed that $(c, \ell)$-diversity is a special case of the basic 3D privacy criterion where the knowledge threshold is $(\ell - 2,\ 0,\ 0)$ and the confidence threshold is $c/(c+1)$, for all sensitive values. In other words, $k$-anonymity provides privacy protection against any adversarial knowledge about the identities of $k-2$ individuals, and $(c, \ell)$-diversity provides privacy protection against any adversarial knowledge about $\ell - 2$ sensitive values that an adversary's chosen target individual does not have.

### 2.4.2 Computation of Breach Probabilities

Detailed discussion of how to compute breach probabilities is beyond the scope of this paper. Here, we only provide key ideas. We first note that one has to carefully pick the form of adversarial knowledge (i.e., kind of logic sentence); otherwise, computation of breach probabilities under background knowledge is likely to be infeasible.

$\Pr(t \text{ has } s \,|\, K, D^*)$ is generally computed based on the *random world assumption*. Intuitively, given a release candidate $D^*$, each possible original data $D$ that can produce $D^*$ by applying the sanitization mechanism to $D$ is called a *possible world* of $D^*$. One commonly used assumption that simplifies probability computation is that, without the given adversarial knowledge, each possible world is equally likely. Notice that "$t$ has $s$" and $K$ are logic sentences that can be evaluated on each possible world and return either true or false. Let $n(X \,|\, D^*)$ denote the number of possible worlds of $D^*$ on which logic sentence $X$ is true. By the definition of conditional probability, we obtain:

$$\Pr(t \text{ has } s \,|\, K, D^*) = \frac{n((t \text{ has } s) \text{ and } K \,|\, D^*)}{n(K \,|\, D^*)}.$$

To compute $\max_{t \in T,\ s \in S,\ K \in \mathcal{L}} \Pr(t \text{ has } s \,|\, K, D^*)$, where $\mathcal{L}$ is a language, it would be computationally infeasible if we try all possible ($t$, $s$, $K$) triples to find the maximum. The trick is to analyze the necessary conditions of the maximum; i.e., find a small set of ($t$, $s$, $K$) triples that includes the maximum solution. Then, restrict the search to that set of ($t$, $s$, $K$) triples. If this restricted set is significantly smaller than the set of all possible ($t$, $s$, $K$) triples, we can observe significant efficiency improvement. After having this restricted set, Martin et al. used dynamic programming to search for the maximum [173]. Chen et al. further proposed a *congregation* property (saying when the breach probability is maximized, all the individuals involved in adversarial knowledge $K$ are in at most two equivalence classes) and showed that their language $\mathcal{L}_{t,s}(\ell, k, m)$ satisfies the property. Based on the congregation property, they improved efficiency over dynamic programming by several orders of magnitude [51].

## 2.5 Protection Against Probabilistic Background Knowledge

Until now, we have described privacy definitions based on adversaries with only precise knowledge; $\ell$-diversity [166, 168] guards against negation statements of the form "Bob does not have heart disease", and in addition Martin et al. [173] and Chen et al. [51] proposed algorithms to guard against implications of the form "If Bob has the flu then Clara has the flu". However, as described in the case of Bayes-Optimal privacy (in Section 2.3) adversaries may possess probabilistic knowledge about parts of the domain. For instance, an adversary may know that the incidence of cancer in Gotham City is only 10%, but is higher (about 50%) if only males in Gotham City are considered. In order to capture such kinds of adversarial knowledge, Evfimievski et al. [92] proposed a privacy criterion called $(\alpha, \beta)$-privacy.

Consider an anonymization algorithm $R$ with input domain $D_U$ and output domain $D_V$. Suppose $R$ acts on a (secret) data item $u \in D_U$ and outputs $v \in D_V$. For example, $R$ may add some random noise into $u$ to generate $v$. Evfimievski et al. say that $R$ allows privacy breaches if for some property $\phi$ about $u$, the adversary's prior probability that $\phi(u) = true$ is very different from the adversary's posterior probability that $\phi(u) = true$ after seeing the output $v$. The adversary's background knowledge is captured in terms of the prior probability, and additional information due to the access to $v$ represented by the posterior probability.

---

**Definition 2.5 ($(\alpha, \beta)$-Privacy).** Let $R$ be an algorithm that takes as input $u \in D_U$ and outputs $v \in D_V$. $R$ is said to allow an *upward* $(\alpha, \beta)$*-privacy breach* with respect to a predicate $\phi$ if for some probability distribution $f$,

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

Similarly, $R$ is said to allow a *downward* $(\alpha, \beta)$*-privacy breach* with respect to a predicate $\phi$ if for some probability distribution $f$,

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

$R$ is said to satisfy $(\alpha, \beta)$-*privacy* if it does not allow any $(\alpha, \beta)$-privacy breach for any predicate $\phi$.

---

Notice that, unlike the privacy criteria in previous sections which define whether or not a release candidate is safe, $(\alpha, \beta)$-privacy defines whether an *anonymization algorithm* is safe. Specifically, $(\alpha, \beta)$-privacy considers all possible inputs (no matter what the data publisher's original data set is) and all possible outputs (no matter what release candidate is actually published) of an anonymization algorithm. If there is an input–output pair that allows a privacy breach, then the anonymization algorithm is not safe.

Evfimievski et al. derived the necessary and sufficient conditions for $R$ to satisfy $(\alpha, \beta)$-privacy *for any prior distribution and any property* $\phi$ in terms of the *amplification* of $R$. An algorithm $R$ is defined to be $\gamma$-*amplifying* if

$$\forall v \in D_V,\ \forall u_1, u_2 \in D_U,\ \ \frac{P(R(u_1) = v)}{P(R(u_2) = v)} \leq \gamma, \tag{2.2}$$

where the probabilities are measured using the random coins of the algorithm $R$.

---

**Theorem 2.2 (From [92]).** Let $R$ be an algorithm that is $\gamma$-amplifying. $R$ does not permit an $(\alpha, \beta)$-privacy breach *for any adversarial prior distribution* if and only if

$$\gamma \leq \frac{\beta}{\alpha} \cdot \frac{1 - \alpha}{1 - \beta} \tag{2.3}$$

---

Unlike the previous privacy definitions, the $(\alpha, \beta)$ condition does not limit the information known to the adversary as it considers every possible adversarial prior belief. Consequently, the anonymization algorithm is forced to satisfy the strict $\gamma$-amplification condition. For instance, no deterministic algorithm (which includes generalization and bucketization schemes) can satisfy $(\alpha, \beta)$-privacy, unless $R$ maps all the inputs to the same output. If $R$ deterministically maps two inputs $u_1$ and $u_2$ to two distinct outputs $v_1$ and $v_2$, its amplification is

$$\frac{P(R(u_1) = v_1)}{P(R(u_2) = v_1)} = \frac{P(R(u_1) = v_1)}{0} = \infty$$

We will describe random perturbation-based techniques that satisfy $(\alpha, \beta)$-privacy in Section 4.2.

## 2.6 Differential Privacy

Organizations are primarily interested in publishing information collected from individuals in the form of relational tables. Each individual contributes to only a few (say, at most $c$) tuples in the table. The differential privacy criterion, proposed by Dwork [85], is designed to guarantee the privacy of individuals and is motivated by the following intuition. The sanitization process should guarantee that, for any individual $i$, the sanitized output generated by including $i$'s data should be nearly indistinguishable from that generated without $i$'s data. In other words, an individual's privacy is guaranteed if given access to the sanitized data set and information about all but one individual, say $i$, in the table, an adversary cannot determine the value of individual $i$'s tuple. For instance, in Table 1.1, even if the adversary knows the disease of all the individuals except Bruce, given access to the sanitized table, the adversary should not be able to say whether Bruce has the heart disease or the flu or even hepatitis.

Similar to $(\alpha, \beta)$-privacy, differential privacy defines whether or not an anonymization algorithm is safe over all possible inputs and outputs. Let $Tup^n$ denote the set of all possible tables having $n$ tuples.

---

**Definition 2.6 ($(c, \epsilon)$-Differential Privacy [85]).** An algorithm $A$ that takes as input a table $T \in Tup^n$ satisfies $(c, \epsilon)$-*differential privacy* if for every output $S$, and every pair of input tables $T_1$ and $T_2$ that differ in at most $c$ tuples,

$$\frac{P(A(T_1) = S)}{P(A(T_2) = S)} \leq \epsilon,$$

where the probabilities are measured using the random coins in algorithm $A$.

---

Differential privacy can be formally motivated in many ways; we present one in terms of $(\alpha, \beta)$-privacy. When considering relational

tables, one can relax the $(\alpha, \beta)$-privacy condition by considering only properties $\phi$ that pertain to individuals as follows.

Suppose $c = 1$ like in Table 1.1; i.e., every tuple in the relation contains the information about a unique individual. First, it is sufficient to guarantee that the adversary's prior and posterior are not very different for individual properties of the form *"Does Bob have cancer"* or *"Does Bob earn more than $50,000"*. These properties can be captured using the set of all functions $\phi$ whose domain is $Tup$ (the domain from which each tuple is picked) and whose range is $\{0, 1\}$; each predicate $\phi(t) = 1$ represents a unique property.

Next, since we are only interested in safeguarding individual properties, we can relax the adversarial knowledge too. Assume, unlike in the case of Martin et al. or Chen et al., that the adversary does not know any information linking two individuals in the table. Now in the worst case, an adversary may know the exact information about all the tuples in the table, except one (the individual being the adversary's target). Moreover, the adversary may know an arbitrary probability distribution $f$ for the target tuple. Dwork et al. [87] term such an adversary as *informed*. The following definition (similar to the semantic privacy in [87]) rephrases the $(\alpha, \beta)$-privacy criterion with only individual properties and the "all but one" adversary model.

---

**Definition 2.7 ($(\alpha, \beta)$-Individual Privacy).** Consider an algorithm $A$ that takes a secret table $T \in Tup^n$ and outputs $S$. $A$ is defined to satisfy $(\alpha, \beta)$-individual privacy against an informed adversary if for every $D \in Tup^{n-1}$, denoting the exact information about $n - 1$ tuples in the table, for every function $\phi : Tup \to \{0, 1\}$ and every probability distribution $f$ on the rest tuple,

$$P_f(\phi(t) = 1|D) \leq \alpha \rightarrow P_f(\phi(t) = 1|D, S) \leq \beta \tag{2.4}$$

$$\text{and } P_f(\phi(t) = 1|D) \geq \alpha \rightarrow P_f(\phi(t) = 1|D, S) \geq \beta \tag{2.5}$$

---

We can use Theorem 2.2 to derive the necessary and sufficient conditions for an algorithm to satisfy Definition 2.7. Note that the adversary already knows the exact values of all but one of the tuples. Hence, it

is enough to consider the amplification based on two tables $T_1$ and $T_2$ that differ in only one tuple. For every such pair of $T_1$, $T_2$ and for every output $S$, we need

$$\frac{P(A(T_1) = S)}{P(A(T_2) = S)} \leq \frac{\beta}{\alpha}\frac{1-\alpha}{1-\beta} \tag{2.6}$$

Now suppose we want the prior and posterior probabilities to always be within a factor of $\epsilon$, for some $\epsilon > 1$. That is, we want $(\alpha, \epsilon \cdot \alpha)$-individual privacy for every value of $\alpha$ between 0 and 1. This would force Equation (2.6) to become,

$$\begin{aligned} \forall\alpha, \frac{P(A(T_1) = S)}{P(A(T_2) = S)} &\leq \frac{\epsilon \cdot \alpha}{\alpha}\frac{1-\alpha}{1-\epsilon \cdot \alpha} \\ \text{iff, } \forall\alpha, \frac{P(A(T_1) = S)}{P(A(T_2) = S)} &\leq \epsilon\frac{1-\alpha}{1-\epsilon \cdot \alpha} \\ \text{iff, } \frac{P(A(T_1) = S)}{P(A(T_2) = S)} &\leq \epsilon \end{aligned} \tag{2.7}$$

Equation (2.7) corresponds to $(1, \epsilon)$-differential privacy condition.

In summary, like $(\alpha, \beta)$-privacy, differential privacy defines whether an *anonymization algorithm* is safe, and not whether a specific release candidate is safe. Intuitively $\epsilon$-differentially privacy is guaranteed if an adversary cannot sufficiently distinguish any two input tables that differ in the data for a single individual based on any output of the algorithm; the $\epsilon$ parameter denotes the extent to which an adversary can distinguish the tables. Semantically, $\epsilon$-differential privacy is stronger than $(\alpha, \beta)$-privacy, since the latter only considers adversarial knowledge about a single individual, but the former considers adversarial knowledge about all individuals in the table. However, by adding adversarial knowledge of exact information about "all but one" individuals in the table, we showed that the variant $(\alpha, \epsilon\alpha)$-individual privacy (for all $\alpha$) is equivalent to $\epsilon$-differential privacy.

## 2.7 Perfect Privacy

Until now we have considered privacy definitions that bound the disclosure of information sensitive to individuals. However, some data are

so secret that an individual may not want *any* information to be disclosed. Such a stringent privacy requirement is termed *perfect privacy* and is equivalent to Shannon's notion of perfect secrecy [232]. More formally, suppose the absolutely sensitive information is captured by the answer to query $Q_S$ over a relational database. Then, publishing a view $V$ (by answering query $Q_V$) of a relational table $T$ violates perfect privacy if for some prior probability distribution $f$ over the domain of all relational tables, and for some answer $S$ to the secret query $Q_S$,

$$P_f(Q_S(T) = S) \neq P_f(Q_S(T) = S \,|\, Q_V(T) = V) \tag{2.8}$$

Again, this means that there is an adversary with prior background knowledge that is captured by the probability distribution $f$, and whose belief about the answer to the secret query changes after seeing the published view.

Unfortunately, it can be shown that for any non-trivial query $Q_S$, publishing any view $V$ violates Shannon secrecy. For example, consider Table 1.1. Suppose Bruce does not want researchers from Gotham City University to learn any information about his disease. So the sensitive query here is

SELECT Disease FROM Hospital WHERE Name = Bruce.

Intuitively, one might expect that publishing the disease information only about women in the hospital would not leak any information about Bruce (who is male). However, there may be some adversary who knows the information that only one of Cary and Bruce has the flu. Thus publishing the information that Cary has the flu leaks the information that Bruce does not have the flu. This changes the adversary's belief about Bruce's disease, thus violating Shannon's secrecy.

Since in most cases sensitive information about one individual does not depend on other individuals, Miklau and Suciu [179] proposed a perfect privacy definition that only guards against adversaries who consider different tuples in a relation to be *independent* of each other. More formally, let $f$ be a probability distribution over all the possible tuples $f : Tup \rightarrow [0, 1]$, where $f(t)$ denotes the probability that tuple $t$ occurs in a database instance. Hence, the probability of a table $T$ is given by

the following product:

$$P_f(T) = \prod_{t \in T} f(t) \times \prod_{t' \notin T} (1 - f(t'))$$

Let $\mathcal{T}$ denote the set of all possible relational tables. Miklau et al. call the pair $(Tup, f)$ a *dictionary*, which defines a probabilistic table (i.e., a probability distribution over the set $\mathcal{T}$ of all possible realizations of such a table). Let $P_f(Q_S = S)$ denote the probability that the query $Q_S$ outputs $S$; i.e.,

$$P_f(Q_S = S) = \sum_{T \in \mathcal{T} \,:\, Q_S(T) = S} P_f(T)$$

---

**Definition 2.8 (Perfect Privacy [179]).** The query $Q_S$ is perfectly private with respect to query $Q_V$ if for every probability distribution $f$ (which considers different tuples to be independent) and for all answers $S$, $V$ to the queries $Q_S$, $Q_V$, respectively,

$$P_f(Q_S = s) \;=\; P_f(Q_S = S \,|\, Q_V = V). \tag{2.9}$$

---

In other words, the query $Q_S$ is perfectly private with respect to $Q_V$ if the adversary's belief about the answer to $Q_S$ does not change even after seeing the answer to $Q_V$, on all tuple-independent probability distributions.

Miklau and Suciu presented an elegant characterization of the above condition in terms of a logical condition on critical tuples. A tuple $t$ is *critical* to a query $Q$, denoted by $t \in crit(Q)$, if $\exists T \in \mathcal{T},\ Q(T \cup \{t\}) \neq Q(T)$. That is, a tuple is critical to a query if removing it from a table changes the answer to the query.

---

**Theorem 2.3 (Critical Tuple Privacy [179]).** Let $(Tup, f)$ be a dictionary. Two queries $Q_S$ and $Q_V$ are perfectly private with respect to each other if and only if

$$crit(Q_S) \cap crit(Q_V) = \emptyset.$$

---

However, checking whether a tuple is critical was shown to be hard for the second level of the polynomial hierarchy ($\Pi_2^p$-complete[1] in the size of the query) even for conjunctive queries (simple SQL queries without aggregation, recursion, or negation). Subsequently, Machanavajjhala and Gehrke [165] showed that the problem is indeed tractable for sufficiently large subclasses of conjunctive queries.

## 2.8 Other Privacy Definitions

Until now, we have described a variety of definitions of privacy. Many extensions and relaxations of these privacy definitions have been discussed in the literature. These may be broadly classified into extensions of $k$-anonymity, extensions of $\ell$-diversity, relaxations of differential privacy, and relaxations of perfect privacy. We briefly discuss these variants in this section.

### 2.8.1 Extensions of $k$-Anonymity

$k$-Anonymity was proposed for deterministic sanitization mechanisms (e.g., generalization). Aggarwal extended it to defining privacy for a randomization mechanism [8]. Let $D = \{\mu_1, \ldots, \mu_n\}$ denote the original data table, where $\mu_i$ is the vector of attributes for the $i$-th individual. Suppose that we want to release a sanitized version $D^*$ of $D$ by adding random noise. For simplicity, suppose each record $\mu_i$ only contains numeric values and we add Gaussian noise. We represent a release candidate $D^*$ by $\{(x_1, \sigma_1^2), \ldots, (x_n, \sigma_n^2)\}$, where $x_i = \mu_i + \epsilon_i$ is the sanitized record of individual $i$, and $\epsilon_i$ is drawn from the multivariate normal distribution with mean 0 and covariance matrix $\sigma_i^2$. Note that $\sigma_i^2$ controls the amount of noise added to the data and needs to be determined by an anonymization algorithm. Let $p(x; \mu_i, \sigma_i^2)$ denote the probability density function of the normal distribution with mean $\mu_i$ and covariance matrix $\sigma_i^2$, which is the probability that the randomized version of the $i$-th record (i.e., $x_i$) takes value $x$.

[1] The first level of the polynomial hierarchy contains the sets NP and co-NP. Problems in $\Pi_2^p$ are conjectured to be harder than problems in NP. For more details we refer the reader to Section 5 in Arora and Barak [23].

Different from a deterministic sanitization mechanism (e.g., generalization), when a randomization mechanism is used, the goal of an anonymization algorithm is to determine the noise levels $\sigma_i^2$, instead of the sanitized records $x_i$, because each $x_i$ is simply generated from a probability distribution defined by $\sigma_i^2$ (and $\mu_i$). Thus, a privacy criterion is also defined on the noise levels $\sigma_i^2$, instead of the actual sanitized records $x_i$.

---

**Definition 2.9 ($k$-Anonymity in expectation).** Release candidate $D^*$ is $k$-anonymous in expectation if, for each individual $i$, the expected number of individuals $j$ whose original records $\mu_j$ are more likely to generate $i$'s sanitized record than $i$'s own original record $\mu_i$ is at least $k$; i.e., for each $i$,

$$\sum_{j=1:j\neq i}^{n} \Pr[p(X_i;\mu_j,\sigma_i^2) > p(X_i;\mu_i,\sigma_i^2)] \geq k, \tag{2.10}$$

where $X_i$ is a normal random variable with mean $\mu_i$ and covariance matrix $\sigma_i^2$, which represents $i$'s sanitized record.

---

To better understand the above definition, let $Y_{ij} \in \{0,1\}$ be a random variable representing whether $\mu_j$ is more likely to generate $X_i$ than $\mu_i$ does. Note that $Y_{ij}$ is a function of $X_i$: $Y_{ij} = 1$ if $p(X_i;\mu_j,\sigma_i^2) > p(X_i;\mu_i,\sigma_i^2)$; otherwise, $Y_{ij} = 0$. Then, $\sum_{j:j\neq i} Y_{ij}$ is the number of individuals whose original record is more likely to generate $i$'s sanitized record $X_i$ than $i$'s own original record is. It can be easily seen that requiring $E[\sum_{j\neq i} Y_{ij}] \geq k$ is equivalent to Formula (2.10).

We note that this definition has several properties that may not be desirable. First, it is possible that the sanitized record $x_i$ is exactly the same as or very close to the original record $\mu_i$. Although the probability that it happens to a given record is small, the probability that it happens to at least one record may be large, especially when the number of records in the data set is large. Note that, when the data publisher does publish an $x_i$ that happens to be the same as $\mu_i$, he can always claim that they are not the same. Although an adversary would not know the fact that they are actually the same, the individual who contributes the

record would not be happy about seeing his record published without any protection. Second, notice that the larger (in terms of the number of records) the original data set is, the smaller the amount of noise is needed. To see this, consider any subset $D_1$ of records of $D$. When we sanitize $D$, the amount of noise that is needed to be added into a record that belongs to the subset $D_1$ is always smaller than the amount of noise that is needed to be added into the same record if we just want to sanitize the subset $D_1$. This property is the consequence of having the summation in Formula (2.10) and it is not clear whether this is desirable. Third, $k$-anonymity in expectation does not guarantee $k$-anonymity with high probability (e.g., probability 0.95). Note that the latter provides a stronger safety guarantee than the former and can be defined by the requirement that, for each $i$, $\Pr\left[\sum_{j\neq i} Y_{ij} \geq k\right] \geq c$, where $c$ is the confidence threshold.

While $k$-anonymity prevents an adversary from precisely identifying individuals' data records, it does not prevent the adversary from knowing that an individual is in the data set. To address this, Nergiz et al. [188] propose $\delta$-presence. Given an external public data table $T$ (that defines the set of the individuals to be considered, e.g., a voter registration list) and two threshold values $\delta = (\delta_{min}, \delta_{max})$, a release candidate $D^*$ of a original data set $D$ is said to satisfy $\delta$-presence if, for any individual $t \in T$,

$$\delta_{min} \leq \Pr(t \in D \,|\, D^*) \leq \delta_{max}.$$

### 2.8.2 Extensions of $\ell$-Diversity

Recall that $\ell$-diversity guarantees privacy (specifically, non-disclosure of a sensitive attribute) by ensuring that within each equivalence class (also called $q^\star$-block, which is a group of tuples with the same generalized value for the quasi-identifier attributes), the $\ell$ most frequent sensitive values have roughly equal proportions. It has been shown that recursive $(c, \ell)$-diversity provides privacy guarantees when the adversary's knowledge is limited to $\ell - 2$ negation statements of the form "Bruce does not have the Flu". Clearly, this formulation does not guarantee privacy in all scenarios. First, an adversary may have background knowledge that cannot be captured by only negation statements; in

Section 2.4 we discussed work by Chen et al. [51] and Martin et al. [173] who formulated more complex forms of background knowledge.

Next, $\ell$-diversity considers sensitive attributes that are categorical, and assumes that the adversary does not know any semantic information about the relationships between attribute values. As an example, consider a 3-diverse table having an equivalence class of 10 individuals that is associated with the three diseases: stomach ulcer (three individuals), dyspepsia (three individuals), and gastroenteritis (four individuals). This table does not allow an adversary to deduce with high probability whether an individual in the equivalence class has one of these three specific diseases. However, the adversary can deduce with certainty that every individual in this equivalence class has a stomach-related disease, which might be considered a breach of privacy. Xiao and Tao [274] proposed a privacy criterion that requires $\ell$-diversity over such general concepts rather than individual values in the sensitive attribute domain. More precisely, consider a hierarchy on the domain of the sensitive attribute, where the leaf nodes are labeled by specific values in the domain of the sensitive attribute, and internal nodes are concepts that generalize all the leaf nodes in its subtree. Figure 2.1 shows one such hierarchy. Xiao and Tao allow users to specify which nodes in the hierarchy are sensitive; i.e., a user could say that the nodes cancer, stomach disease, and heat-related disease are sensitive nodes. For privacy, they require $\ell$-diversity on sensitive nodes that do not have sensitive parents; i.e., in Figure 2.1, $\ell$-diversity is required

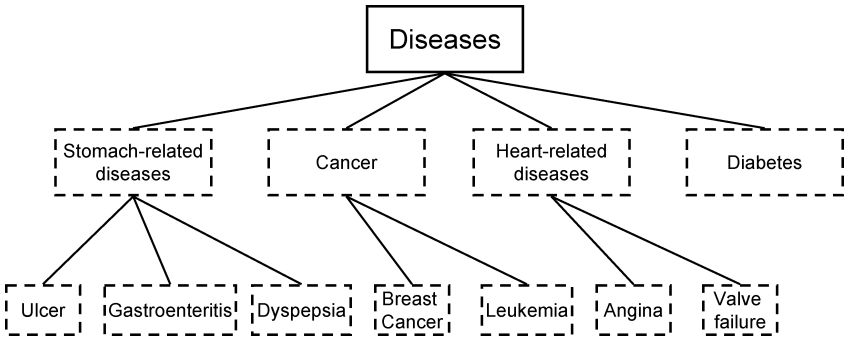



Fig. 2.1 Hierarchy of concepts on the sensitive attribute Disease. Dashed boxes are considered sensitive.

on "stomach-related diseases", "cancer", "heart-related diseases," and "diabetes".

We should note that a variant of $\ell$-diversity (also proposed in [166]) is able to handle some form of semantic information about the degree of sensitivity of a value of a sensitive attribute. In this case, a data publisher decides on a subset $Y$ of values of the sensitive attribute. Values in $Y$ are considered minimally sensitive so that "flu" might be in $Y$ but "AIDS" should not be; a frequently occurring value may also be included in $Y$ at the discretion of the data publisher. Thus $Y$ is called a *don't-care set* [168]. With this set $Y$, a version of $\ell$-diversity that is called *positive disclosure $\ell$-diversity* [166, 168] is designed to protect the sensitive values that are not listed in $Y$. This modified version of $\ell$-diversity addresses charges that were later made by Li et al. [157] that $\ell$-diversity is unnecessary to achieve when some values are much more sensitive than others. It also addresses their second complaint that skewed data such as 49 AIDS patients with one healthy patient in an equivalence class is treated the same as 49 healthy patients and one AIDS patient. This problem is easily solved using recursive $(c, \ell)$-diversity with don't-care sets and a properly chosen constant $c$. For more fine-grained control, each sensitive value could have its own constant $c$, which essentially places a cap on the fraction of times a sensitive value can appear inside an equivalence class. Fine-grained control for each sensitive value was explored in [51].

Yet another set of extensions target privacy for numeric attributes. Here again, ensuring the diversity of specific numeric values may not be enough to guarantee conventional notions of privacy. For example, a 3-diverse table may contain three salaries: 100 K, 101 K, and 99 K, but an adversary can deduce that the salary is between 99 K and 101 K, which appears to be a breach of privacy. Several approaches have been proposed to handling this problem [151, 155, 157]; most recently, Li et al. proposed *proximity-aware privacy*, which requires that every equivalence class contain sensitive values from at least $\ell$ disjoint ranges (of width more than a pre-specified parameter) in roughly equal proportions.

The above papers advance the state-of-the-art by correctly recognizing that semantic information about sensitive attributes is important,

and potentially increases the risk of a privacy breach. However, the work in this space has not yet been able to formalize the range of semantic information that an adversary might possess. For instance, there may not be a single hierarchy of concepts on the sensitive attribute domain; it is thus difficult to reason about the attacker's "mental taxonomy." Similarly, $\ell$-diversity on numeric ranges is implicitly designed to protect against an adversary with background knowledge captured by negation statements of the form *"Bruce's salary is not $100,000"* or *"Bruce's salary is not between $99,000 and $101,000"*. However, it is also reasonable to consider inequality statements like *"Bruce's salary is greater than $100,000"*, which are not captured by the existing notion of proximity-aware privacy. Further work is necessary to precisely characterize the semantics of sensitive attribute values in relationship to extended privacy definitions.

Finally, $\ell$-diversity only ensures that the posterior probability that, say, Bruce has cancer given the published table and the adversarial knowledge is less than $c/(c+1)$. However, in some cases, an adversary may know statistical information that 70% of males in Gotham City above the age of 40 have cancer; here, learning that Bruce has cancer with probability of 75% may not violate his privacy. $t$-Closeness [157] attempts to guarantee privacy against such adversaries by assuming that the distribution of the sensitive attribute (say disease) in the whole table is public information. Privacy is said to be breached when the distribution of the sensitive attribute in an equivalence class is not close to the distribution of the sensitive attribute in the whole table. For instance, Table 2.3 satisfies $t$-closeness, since the disease distribution

Table 2.3. Table satisfying $t$-closeness.

| Non-Sensitive | | Sensitive | |
|---|---|---|---|
| Age | Gender | Disease | Count |
| $< 40$ | $M$ | Flu | 400 |
| $< 40$ | $M$ | Cancer | 200 |
| $\geq 40$ | $M$ | Flu | 400 |
| $\geq 40$ | $M$ | Cancer | 200 |
| $\geq 40$ | $F$ | Flu | 400 |
| $\geq 40$ | $F$ | Cancer | 200 |

in each equivalence class is the same as the disease distribution in the whole table.

While $t$-closeness raises an important point that privacy should be measured relative to the adversary's prior information, its particular implementation is ad hoc. There is no clear characterization of what kind of background knowledge an adversary might have. One could argue that $t$-closeness guards against adversaries who know the marginal distribution of the sensitive attribute in the table being published. This means that $t$-closeness will never say that a table is too sensitive for anything to be published, even if it only includes one individual. Also, when the marginal distribution of the sensitive attribute in the data is very different from the general belief and is indeed sensitive, $t$-closeness may incorrectly assume that the adversary knows that distribution and release it. We believe that a data publisher should not automatically assume that the adversary knows the marginal distribution of the sensitive attribute in the table (a) when the total number of individuals in the table is small, or (b) if there exists some public information about the distribution that is different from the marginal distribution in the table being published.

### 2.8.3 Relaxations of Perfect Privacy

Perfect privacy is breached when the adversary's prior belief about the answer to the secret query changes on seeing some published data. Due to this, Miklau and Suciu showed that no aggregate information relating to the sensitive query can be published; e.g., if "Does Bruce have Cancer" is the sensitive query, then any aggregate that includes Bruce's disease cannot be released. Dalvi et al. [61] propose a relaxation wherein privacy is breached only if the adversary's prior belief *asymptotically* differs from his posterior belief as the size of the database increases to infinity. The authors show that this relaxation allows some aggregates to be published. Stoffel and Studer [237] propose *certain answer privacy* wherein privacy is breached only if the adversary can say that some tuple is certainly in the answer to the secret query, given the published data.

### 2.8.4 Relaxations of Differential Privacy

Variants of differential privacy have been discussed in [49, 86, 167]. The basic idea is the following. Differential privacy requires that, given *any* output of the algorithm, the adversary should not be able to distinguish between any two input tables $D_1$ and $D_2$ that differ in one tuple. However, given $D_1$ and $D_2$, some of the outputs are very unlikely. For instance, let $D_1$ and $D_2$ have 100 tuples; let $D_1$ be a table with all 0s and $D_2$ be a table with one 1 and rest 0s. Let the algorithm be sampling with replacement 100 times. Consider an output that has 100 tuples, all of which are 1. This output is very unlikely (with probability $10^{-200}$). So it might be acceptable to allow a privacy breach with such a low probability. This is the motivation behind probabilistic differential privacy [167], where differential privacy may be violated by a set of outputs whose total probability is less than a small constant $\delta$.

## 2.9 Discussion and Comparison

In this section, we have described several distinct privacy definitions and their extensions. All of these have different intuitions and operate on different adversarial assumptions. While this state-of-the-art may be good in terms of our understanding of the definition of privacy, it poses the following problem to a data publisher: *"Which privacy definition should be used for a specific application?"* Unfortunately, there is neither a mandate on how to define privacy for a new application, nor a clear technique to compare the various privacy definitions prevalent in the literature. The problem can be solved if all privacy definitions can be expressed under one common framework. In this section we describe initial work toward one such unification based on a framework for privacy definitions presented by Machanavajjhala [164].

### 2.9.1 Semantic Privacy Definitions

In order to unify privacy definitions, we must first understand the common denominators underlying the definitions. Every privacy definition must answer three important questions:

- *What information should be kept private?*

- *How is private information leaked; i.e., how does an adversary learn private information?*
- *When does information disclosure lead to a privacy breach, and how is disclosure measured?*

Some privacy definitions, e.g., $(c, k)$-safety, 3D privacy (and the related privacy skyline), $(\alpha, \beta)$-privacy, and perfect privacy, explicitly answer these questions. We follow Machanavajjhala [164] and term these definitions as *semantic privacy definitions*. Other privacy definitions like recursive $(c, \ell)$-diversity, $t$-closeness, and differential privacy do not explicitly state these assumptions; these algorithmic criteria are termed *syntactic definitions*. Some of these syntactic criteria like $\ell$-diversity and differential privacy are equivalent to semantic variants that explicitly state the assumptions. Below we summarize all the privacy definitions we described in terms of their semantic variants.

Any semantic privacy definition has the following structure. The sensitive information can be described in terms of a set of sensitive predicates. For instance, in $(c, k)$-safety the truth value of every predicate of the form "Bruce has Cancer" is sensitive. In perfect privacy, the truth value of the predicate "$S$ is the answer to $Q_S$" is considered sensitive for every $S$. Next, the adversarial background knowledge can be described by a set of adversarial distributions on the space of input tables, and that is *independent* of any specific input table. Again, in the case of $(c, k)$-safety, each adversarial distribution is captured by a Boolean formula over statements about the sensitive attribute of individuals in the population, and all adversarial distributions that correspond to Boolean formulas expressed by at most $k$ implications are considered. Finally, the privacy metric is described in terms of the prior and posterior probabilities in each adversarial distribution. In $(c, k)$-safety, privacy is breached if the posterior probability is greater than or equal to $c$ for some adversarial distribution that is considered.

***k*-Anonymity** [241]:

- Sensitive information: For every individual $i$, the actual association of the record of $i$ to the identity of $i$ is sensitive. Let the record ID attribute (which contains identifiers or keys,

each of which uniquely identifies one record) be denoted by $RID$. A predicate of the form "$i[RID] = n$" (e.g., "This particular record belongs to Bob") is sensitive.
- Adversarial background knowledge/belief: All input tables are equally likely. Furthermore, the adversary may know a conjunction of $k-2$ statements of the form $i[RID] = n'$ (denote the set of all such conjunctions by $\mathcal{L}_{pos}(k-2)$).
- Privacy metric: Publishing $D^\star$ breaches privacy if $\exists i, \exists n \in RID, \exists K \in \mathcal{L}_{pos}(k-2)$ such that

$$P(i[RID] = n \,|\, K, \text{ published data } D^\star) = 1.$$

**$\ell$-Diversity** [166, 168]:

- Sensitive information: For every individual $i$ and every sensitive value $s \in S$, the predicate "$i[S] = s$" is sensitive. E.g., "Does Bob have cancer?"
- Adversarial background knowledge/belief: All input tables are equally likely. Furthermore, the adversary may know a conjunction of $\ell - 2$ negation statements of the form $i[S] \neq s'$ (denote the set of all such conjunctions by $\mathcal{L}_{neg}(\ell-2)$).
- Privacy metric: Publishing $D^\star$ breaches privacy if $\exists i, \exists s \in S, \exists K \in \mathcal{L}_{neg}(\ell-2)$ such that

$$P(i[S] = s \,|\, K, \text{ published data } D^\star) \geq c/(c+1).$$

**($c$,$k$)-Safety** [173]:

- Sensitive information: For every individual $i$ and every sensitive value $s \in S$, the truth value of predicate "$i[S] = s$" is sensitive. E.g., "Does Bob have cancer?"
- Adversarial background knowledge/belief: All input tables are equally likely. Furthermore, the adversary may know a Boolean formula that can be expressed as a conjunction of $k$ implication statements (denote the set of all such conjunctions by $\mathcal{L}_{basic}(k)$).

- Privacy metric: Publishing $D^\star$ breaches privacy if $\exists i, \exists s \in S, \exists K \in \mathcal{L}_{basic}(k)$ such that

$$P(i[S] = s \,|\, K, \text{ published data } D^\star) \geq c.$$

**3D Privacy Criterion** [51]:

- Sensitive information: For every individual $i$ and every sensitive value $s \in S$, the truth value of predicate "$i[S] = s$" is sensitive. E.g., "Does Bob have cancer?"
- Adversarial background knowledge/belief: All input tables are equally likely. Furthermore, the adversary may know (1) $\ell$ sensitive values an individual does not have, (2) the sensitive values of $k$ other individuals, and (3) an implication of the form: if one of $i_1, \ldots, i_m$ have sensitive value $s$, then $i$ has sensitive value $s$ (denote the set of all such conjunctions of (1), (2), and (3) by $\mathcal{L}_{i,s}(k, \ell, m)$).
- Privacy metric: Publishing $D^\star$ breaches privacy if $\exists i, \exists s \in S, \exists K \in \mathcal{L}_{i,s}(k, \ell, m)$ such that

$$P(i[S] = s \,|\, K, \text{ published data } D^\star) \geq c.$$

**$(\alpha, \beta)$-Privacy and $\gamma$-Amplification** [92]:

- Sensitive information: For every individual $i$, let $D$ be the domain of $i$'s tuple. For every $D' \subseteq D$, the truth value of predicate "$i \in D'$" is sensitive. E.g., "Does Bob have one of cancer, heart disease, or the Flu?", or "Does Bob not have ulcer?"
- Adversarial background knowledge/belief: An arbitrary probability distribution over the space of input tables.
- Privacy metric: An anonymization algorithm allows privacy breaches if $\exists i$, $\exists D' \subseteq D$, $\exists$ output table $D^\star$, $\exists$ probability distribution $P$ such that

$$P(i \in D') < \alpha \;\wedge\; P(i \in D' \,|\, D^\star) > \beta.$$

**$\epsilon$-Differential privacy** [87, 85]:

- Sensitive information: For every individual $i$, let $D$ be the domain of $i$'s tuple. For every $D' \subseteq D$, the truth value of predicate "$i \in D'$" is sensitive. E.g., "Does Bob have one of cancer, heart disease, or the Flu?", or "Does Bob not have ulcer?"
- Adversarial background knowledge: The adversary knows exact information about all individuals in the table except for record $i$, and the adversary also has an arbitrary probability distribution $P$ over the value of record $i$. Let $A$ denote such an *informed* adversary.
- Privacy metric: An anonymization algorithm allows privacy breaches if $\exists i, \exists D' \subseteq D, \exists$ output table $T^\star, \exists$ informed adversary $A$ such that

$$\frac{P_A(i \in D' \,|\, T^\star)}{P_A(i \in D')} > \epsilon.$$

**Perfect Privacy** [179]

- Sensitive information: Given a secret query $Q_S$, then for every possible answer $S$ to $Q_S$, the predicate "$S$ is the answer to $Q_S$" is sensitive. E.g., if $Q_S$ is the query "Names of co-authors of Ashwin", then the answer to the question "Is Dan Ashwin's co-author" is sensitive.
- Adversarial background knowledge: The adversary knows an arbitrary probability distribution $P$ over tuples in the table. All tuples in a database instance are considered independent.
- Privacy metric: Answering query $Q_V$ breaches privacy if $\exists S, \exists$ published view $V, \exists$ tuple-independent probability distribution $P$ such that

$$P(Q_S = S | Q_V = V) \neq P(Q_S = S).$$

Some other privacy definitions may look like semantic definitions, but are not. For example, $t$-closeness does not have an equivalent

semantic definition. $t$-Closeness can be seemingly rephrased in terms of a semantic definition as follows:

- Sensitive information: For every individual $i$ and every sensitive value $s \in S$, the truth value of predicate "$i[S] = s$" is sensitive. E.g., "Does Bob have cancer?"
- Adversarial background knowledge: All input tables are equally likely. Furthermore, the adversary knows the exact marginal distribution of the sensitive attribute in the data that are input to the anonymization algorithm.
- Privacy metric: Privacy is breached if the distribution $P(i[S] \,|\, K$, published data $D^\star)$ is not close to the marginal distribution of the sensitive attribute in $D^\star$ (measured by a distance function).

However, the above definition is *not* a semantic definition because the adversary's prior knowledge *depends* on the specific database being sanitized. The above list of semantic definitions also point out a significant difference of $t$-closeness from the rest of the privacy definitions. While other privacy definitions guarantee privacy under a set of adversarial distributions, $t$-closeness guarantees privacy against a single adversary. This is because $t$-closeness assumes that the data publisher exactly knows what the adversary know, while this is not true in other privacy definitions.

### 2.9.2 Publishing Multiple Horizontal Partitions

In some data publishing scenarios, data are collected periodically and also published periodically. Different privacy definitions behave differently in the scenario of multiple or periodic releases. In this section, we consider a simple case of multiple releases, namely multiple releases of horizontal partitions of a table, and point out a subtle issue for some privacy definitions in this case. A general discussion of multiple releases is in Section 7.4.2.

We illustrate the issue by applying $t$-closeness to multiple releases of Table 2.3 in a straightforward way. If the data publisher wants to publish the entire Table 2.3, then by the assumptions of $t$-closeness, the

data publisher would assume that the adversary knows that the ratio of Flu:Cancer is 2:1 over all the individuals in the table. Now, suppose the data publisher acquires the data in Table 2.3 piece by piece: first the information on males is collected and then the information on females is collected. The data publisher also releases sanitized versions piece by piece: first the information about males and then the information about females. Note that one might believe the assumption that each piece is independent of the other is reasonable because they share no common individuals and their domains are disjoint. Thus it may seem that there should not be any issues with multiple release of data over time. However, in this scenario the data publisher would assume that the adversary knows the distribution of sensitive values among males (when publishing the first piece) and also the distribution of sensitive values among females (when publishing the second piece). This is more knowledge than if the both pieces had been released as one table and could lead to trouble as follows.

In the extreme case, consider two data publishers $A$ and $B$. Suppose $A$ decides to publish Table 2.4 and $B$ publishes each equivalence class in Table 2.4 in a separate table (i.e., $B$ publishes four tables, the disease distribution of males under 20, the disease distribution of males between 20 and 40, the disease distribution of males above 40, and the disease distribution of females above 40). According to $t$-closeness, $A$ should not publish the data while $B$ should. Data publisher $A$ should not publish the data since the distribution of cancer among males between 20 and 40 years of age is very different from the overall distribution of cancer (in fact, $A$ should publish Table 2.3 instead). On the other hand, $B$ is allowed to publish the four tables, since the $t$-closeness assumption

Table 2.4. 200-Anonymous table.

| Non-Sensitive | | Sensitive | |
|---|---|---|---|
| Age | Gender | Disease | Count |
| $< 20$ | $M$ | Flu | 400 |
| $20 - 40$ | $M$ | Cancer | 200 |
| $\geq 40$ | $M$ | Flu | 400 |
| $\geq 40$ | $M$ | Cancer | 200 |
| $\geq 40$ | $F$ | Flu | 400 |
| $\geq 40$ | $F$ | Cancer | 200 |

lets $B$ assume that the adversary already knows the distribution of the sensitive attribute in each table. However, $A$ and $B$ were indeed considering publishing the same information, thus leading to an apparent inconsistency.

The solution to this dilemma is to realize that $t$-closeness assumes that the tuples in the data are *dependent*. Two equivalence classes are, in fact, not independent once we assume the adversary already knows the overall distribution of sensitive values since the two equivalence classes contribute to this distribution. Privacy definitions like $(c, k)$-safety and 3D privacy (and its more general version called skyline privacy) also assume tuples are correlated (because the form of background knowledge that an adversary may have included relationships between individuals across equivalence classes). Thus for these definitions one has to be careful about horizontally partitioning the data and then reasoning about each partition separately. One may even have to reason about the effect that future data may have (e.g., should we assume the adversary knows the distribution of the sensitive attribute among the males in the current sample, in which case this sample is independent of future data; or should we assume the adversary knows what the distribution of the sensitive attribute will be once we also collect data on female patients, in which case the sample is dependent on future data?). Without careful consideration of this issue, the privacy guarantee might end up being inconsistent; previously released safe tables may become unsafe because of a new release that is safe on its own. Thus, the publication of horizontal partitions of the data over time is an important research problem for such privacy definitions. We will discuss issues of other kinds of multiple releases in Section 7.4.2.

### 2.9.3 Summary

Different applications will need to keep different kinds of information secret, and they will require different assumptions about the adversary. For instance, a military application might require absolutely no information to be leaked about some queries, forcing the use of perfect privacy. Some applications may need to model adversaries using complex distributions; in this case, one may want to use differential privacy

or $(\alpha, \beta)$-privacy. In other applications, simpler adversary models and generalization schemes might be sufficient, and $\ell$-diversity, $(c, k)$-safety, 3D privacy criterion, and their variants might be sufficient. We believe that it is better for data publishers to choose privacy definitions that are themselves semantic (or provably equivalent to some semantic definition); this helps to evaluate whether the definitions suit the needs of the application, and allows the definitions to be adapted to new applications. Definitions that do not have a semantic equivalent may result in subtle and unanticipated problems.

# 3

# Utility Metrics

> The temptation to form premature theories upon insufficient data is the bane of our profession.
>
> — Sherlock Holmes (Sir Arthur Conan Doyle)

A data publisher seeks to release data that are not only safe, but also useful. In this section, we discuss ways of measuring the amount of useful information that is still in the data after sanitization. These measures are needed by a data publisher to evaluate the utility of different release candidates, and they are also needed by the data recipient to gauge how useful an analysis will be.

Collections of data about individuals provide two kinds of information. The first kind is individual-level personal information. Consider the medical record example. This is the kind of information a doctor would need to treat a particular patient. It is also the kind of information an attacker would need to run a convincing scam. The second kind of information is statistical information about a population. This is the kind of information that is of interest to medical researchers and economists, and the kind of information that a data publisher wants to publicize. When sanitizing a data set, some instance-level

sensitive information is invariably removed. However, the unintended consequence is that some aggregate or statistical information is also lost. Thus, the *utility* of a sanitized data set is intuitively measured by the extent to which it preserve aggregate and statistical information.

In general, there are two ways to evaluate the quality of sanitized data. The first approach is to actually use the data as input to a query or an analysis task, and to evaluate the quality of the results. We postpone discussion of this approach to Section 5. The second approach, described in this section, is to develop one or more quantitative measures of information loss, which an anonymization algorithm could try to optimize. Recent results indicate that this optimization approach should be used with caution. The first reason is that a poorly chosen measure of information loss could degrade the quality of the sanitized data. For example, recent work by Nergiz and Clifton [190] has shown experimentally that if the goal is to build a good classifier from sanitized data, then optimizing for the LM, DM, CM, or AM metrics (discussed in the following sections) may provide little benefit. The second reason for caution is that in certain cases, the act of optimizing an information loss measure subject to privacy constraints can itself leak additional information [94, 271] (for more details, see Section 6). With these caveats in mind, let us discuss some proposed measures of utility.

Many utility measures have been proposed in the literature. Rather than provide a laundry list of formulas, we will discuss a smaller set that illustrates the key ideas that are in use.

**Generalization/Suppression Counting:** One of the earliest and most intuitive measures of information loss involves counting the number of anonymization operations performed on a data set. For example, one of the key operations in the $k$-anonymity framework is *generalization*, which coarsens the value of an attribute (e.g., changing "age $=$ 20" to "age $\in [10 - 30]$"). If generalization is the only operation being performed, then it is reasonable to measure information loss by the number of generalization steps performed. Samarati used one version called *generalization height* [226]. In their proof of the NP-hardness of $k$-anonymity, Meyerson and Williams used another variation: they measured the total number of attribute values that were suppressed

[178]. One can even imagine weighted version of these methods since some attributes may be more important than others.

One problem with this approach is that not all operations affect utility in the same way. A generalization operation that maps "male" to "*" and "female" to "*" effectively removes gender information from the data while a generalization operation that turns age into an age range of length 2 (i.e., $[0-1], [2-3]$, etc.) seems much more benign. Iyengar [132] addresses these issues with two measures of information loss: the aptly named *loss metric* (LM) and the *classification metric* (CM).

**Loss Metric (LM):** LM is defined in terms of a normalized loss for each attribute of every tuple. For a tuple $t$ and categorical attribute $A$, suppose the value of $t[A]$ has been generalized to $x$. Letting $|A|$ represent the size of the domain of attribute $A$ and letting $M$ represent the number of values in this domain that could have been generalized to $x$, then the loss for $t[A]$ is $(M-1)/(|A|-1)$. The loss for attribute $A$ is defined as the average of the loss $t[A]$ for all tuples $t$. The LM for the entire data set is defined as the sum of the losses for each attribute.

**Classification Metric (CM):** The classification metric (CM) is designed to measure the effect of the anonymization on a hypothetical classifier. In this scenario, there is a distinguished class attribute, and tuples are placed into groups (usually they are grouped by quasi-identifier value). Each tuple incurs a penalty of 1 if it is suppressed or if its class attribute is not the same as the majority class attribute in the group. The classification metric is defined as the average of the penalties of all the tuples. Similar ideas were presented by Wang et al. [259] and LeFevre et al. [151] as local measures of information loss to guide anonymization algorithms.

**Discernibility Metric (DM):** Bayardo and Agrawal [29] proposed a metric similar in spirit to Iyengar's LM called the *discernibility metric* (DM). DM assigns a penalty to each tuple based on how many other tuples in the database are indistinguishable from it, and therefore it works naturally in the $k$-anonymity framework. For a database of size $n$, DM assigns a penalty of $n$ for each suppressed tuple. If a tuple is not suppressed, the penalty it receives is the total number of tuples in the database having the same quasi-identifier values. Thus, when tuples are

grouped by quasi-identifier, the DM for a data set is the sum of squared group sizes plus $n$ times the number of suppressed tuples. Average group size (instead of sum of squared group sizes) has also been used [166, 150].

**Ambiguity Metric (AM):** Nergiz and Clifton [190] proposed another metric, called the *ambiguity metric* (AM), that is especially suitable for the $k$-anonymity framework. For each tuple $t^*$ in the sanitized data, AM considers the number of tuples in the *domain* of the data that could have been mapped (generalized) to $t^*$. This number is the ambiguity of $t^*$. The AM for the sanitized data set is then the average ambiguity for all tuples in the sanitized data.

**KL-Divergence:** Most of the metrics discussed thus far are oblivious to the distribution of actual attribute values in the data. If age was uniformly distributed, and independent of all other attributes, then replacing the age attribute with an age range would have little effect since a data analyst is very likely to take the age range and, following the principle of maximum entropy, assume a uniform distribution of ages within the range. In this case the analyst's assumption is accurate. On the other hand, if the age distribution were skewed, then the uniformity assumption could bias the analyst's results. For this reason, a utility metric commonly used in the statistics community and known as KL-divergence would be more appropriate for measuring the information loss of sanitized data [75, 142]. To use KL-divergence, the original table is treated as a probability distribution $p_1$ as follows. $p_1(t)$ is the fraction of tuples equal to $t$. The sanitized data are also converted to a probability distribution $p_2$ (possible ways of doing this will be discussed next). The KL-divergence between the two is $\sum_t p_1(t) \log \frac{p_1(t)}{p_2(t)}$. The larger this number is, the greater the information loss. There are many ways of interpreting the sanitized data as a probability distribution. If the sanitized version of the data is a set of histograms, then the histograms can be interpreted as constraints and the probability distribution $p_2$ is the maximum entropy distribution consistent with those constraints. Another way is to posit a statistical model such that the sanitized data form the *sufficient statistics* [229] of the model. One example of this is the fact that histograms (also known as marginals)

form the sufficient statistics for a class of models known as loglinear models [53]. When this approach is taken, then the KL-divergence has the following nice interpretation. A model that overfits the original data (i.e., a multinomial model with one parameter for every possible tuple) has the maximum likelihood $L_1$ on this data set. A model using the sanitized data as sufficient statistics has lower likelihood $L_2$. The quantity $\log \frac{L_1}{L_2}$ is known as the log-likelihood ratio and it measures the amount of likelihood that is *not* captured by the model built from sanitized data (thus a value of 0 means that all of the likelihood is captured by such a model). It is well-known that the log-likelihood ratio is formally equivalent to KL-divergence.

Another information theoretic metric was proposed by Gionis and Tassa [115] and is applicable to anonymization algorithms that partition the data into groups. It is computed as the sum of the entropies of each attribute for each group. Xu and Ye [277] proposed the use of the difference in entropy of the quasi-identifier between the original data and the sanitized data. The overall change in entropy was proposed by Gomatam and Karr [116], and for uses of conditional entropy see [264].

**$L_p$ norm:** KL-divergence is not the only way to measure the distance between the original probability distribution and the probability distribution reconstructed from the sanitized data. Agrawal and Aggarwal [15] used the $L_1$ norm. The $L_1$ norm is an example of an $L_p$ norm, which is defined as $\{\sum_t |p_1(t) - p_2(t)|^p\}^{1/p}$ for $p < \infty$ and $\max_t |p_1(t) - p_2(t)|$ for $p = \infty$. Any $L_p$ norm (where $p \geq 1$) can also be used to measure the distance between the original and reconstructed probability distributions. The *total variation* distance is equal to one-half of the $L_1$ norm. For numeric data it can make more sense to estimate the original values (from the sanitized data) instead of directly computing probabilities. This approach is used for sanitized streams [154, 198] where the $L_2$ norm between the original and reconstructed stream is used as a measure of the variance still remaining in the sanitized data.

**Hellinger Distance:** Another statistical measure of dissimilarity between distributions is known as Hellinger Distance. It is defined as $\sqrt{\sum_t \left(\sqrt{p_1(t)} - \sqrt{p_2(t)}\right)^2}/2$ and is used in [116].

**Bivariate Measures:** Gomatam and Karr [116] also discuss bivariate measures of information loss. For a pair of attributes $A$ and $B$, they compute the $\chi^2$ statistic in both the original data and the sanitized data. The $\chi^2$ statistic is then used to compute either the Cramer's V or Pearson's contingency coefficient C. The information loss measure is then the difference in Cramer's V (or Pearson's contingency coefficient C) from the original data and sanitized data. For more details, see [116].

**Workload-Aware Metrics:** LeFevre et al. [151] argue that the utility metric should depend on the intended uses of the sanitized data (in cases where the use is known beforehand). The uses considered are classification, regression, and answering count queries over regions specified by range queries. These metrics apply to algorithms that partition the domain of the quasi-identifier into groups. For classification, the goal is to be able to predict the value of a distinguished attribute called the class attribute. The corresponding measure of information loss is the weighted average of the entropy of the class attribute in each group. If there are multiple class attributes, then the total information loss is the sum of the information loss for each attribute. In regression problems, the class attribute is continuous, so the information loss is measured as the weighted average of the variance of the class attribute in each group.

In the case of count queries, the measure of information loss for each query is called *imprecision*. Imprecision is measured as the number of points in all groups that overlap with the selection region of the query minus the true answer. The total imprecision is the sum of the imprecision of all the queries. Zhang et al. [282] measure the information loss in a partition as the difference between the maximum and minimum value of a distinguished numeric attribute (usually the sensitive attribute) in the partition. The total information loss is measured as either the maximum of these losses or the sum of these losses.

**First- and Second-Order Statistics:** Torres [254] used measures of information loss that are minimized when the original data and the sanitized data have the same first- and second-order statistics. Our discussion here follows Sanchez et al. [227], and assumes that there are $p$ attributes (all numeric) and that the original data and sanitized data both have the same number of tuples ($n$).

A variety of different metrics can be defined by comparing descriptive statistics computed on the original data with those computed on the sanitized data. In the following, let $x_{ij}$ denote the value of attribute $j$ for tuple $i$ in the original data, and let $x'_{ij}$ be the corresponding value for the sanitized data. Let $\mu_i$ ($\mu'_i$) denote the mean of attribute $i$ in the original (sanitized) data, and let $v_{ij}$ ($v'_{ij}$) denote the covariance between attributes $i$ and $j$ in the original (sanitized) data. Similarly, let $\rho_{ij}$ and $\rho'_{ij}$ be the correlation between attributes $i$ and $j$ in the original and sanitized data, respectively.

Some sample metrics described in [227] include:

- Assuming a one-to-one map between tuples in the original and sanitize data sets, the *mean variation* is defined as $\frac{1}{np}\sum_{i=1}^{n}\sum_{j=1}^{p}\frac{|x_{ij}-x'_{ij}|}{|x_{ij}|}$.
- The *variation of the means* is defined as $\frac{1}{p}\sum_{i=1}^{p}\frac{|\mu_i-\mu'_i|}{|\mu_i|}$.
- The *variation of covariances* is $\frac{1}{p(p+1)/2}\sum_{i=1}^{p}\sum_{1\leq j\leq i}\frac{|v_{ij}-v'_{ij}|}{|v_{ij}|}$.
- The *variation of variances* is $\frac{1}{p}\sum_{i=1}^{p}\frac{|v_{ii}-v'_{ii}|}{|v_{ii}|}$.
- The *mean absolute error of correlations* is $\frac{1}{p(p-1)/2}$ $\sum_{i=1}^{p}\sum_{1\leq j<i}|\rho_{ij}-\rho'_{ij}|$.

Clearly countless variations of these measures can be produced by replacing relative errors with absolute errors (and vice versa), replacing absolute values with squares of absolute values, etc. These measures can also be combined by taking weighted averages (for more details, see [227]).

**Analytical Validity:** A similar, but less formal, approach can be seen in the statistics literature (see for example [1, 143, 218]). The amount of information present in the sanitized data is known as *analytical validity*, and is evaluated by building models over both the original data and the sanitized data, and then comparing the learned parameters. Usually this is done by computing confidence intervals for the parameters learned from the original data and observing how many times the parameters from the sanitized model fall into the computed confidence intervals. Karr et al. [139] initiate a formal study on this topic by proposing to compute the average probability of overlap in confidence

intervals and the average relative overlap. A utility measure can also be defined without building a model by computing statistics over the original and sanitized data and comparing the results [81].

**Invariance:** The ideas in these workload-aware methods can even be taken to the following extreme. Bu et al. [40] suggest that anonymization schemes should be devised so that they do not alter the output of pre-selected data mining algorithms. This approach would typically apply in situations where data are outsourced for the purposes of data mining. The resulting model (built by an external expert on sanitized data) could then be processed by the data owner to yield the same model that would have been built over the original data.

**Reconstructibility:** The approaches discussed so far measure the utility of the sanitized data that are actually produced. It is also possible to measure utility in terms of the algorithm used to create the sanitized data; in this case, the result is usually a probabilistic utility guarantee.

As one example, Agrawal et al. [17] define the utility associated with a randomized anonymization algorithm in terms of the ability to reconstruct statistics from the sanitized data. More formally, if $f$ is a real-valued function computed over the original data, and $f'$ is the estimator of $f$ computed over the sanitized data, then $f$ is $(n, \epsilon, \delta)$ *reconstructible* if $|f - f'| < \max(\epsilon, \epsilon f)$ with probability at least $(1 - \delta)$ whenever the number of tuples in the original data is at least $n$. Thus utility can be defined in terms of the class of functions that are $(n, \epsilon, \delta)$ reconstructible.

Rastogi et al. [211] measure utility in terms of how likely it is that the answer to a count query will be smaller than the sampling error. Formally, a randomized anonymization algorithm is $(\rho, \epsilon)$-useful if for any count query $Q$, with probability $(1 - \epsilon)$, the absolute difference between the true answer to $Q$ and the estimated answer (computed from sanitized data) is at most $\rho\sqrt{n}$ (where $n$ is the number of tuples in the data).

For anonymizations that can be expressed in terms of matrix multiplication, Agrawal and Haritsa [18] observed that reconstruction accuracy depends on the condition number of the anonymization matrix, and they used this condition number as a measure of information loss.

# 4

# Mechanisms and Algorithms

> An algorithm must be seen to be believed.
>
> — Donald Knuth

Data publishing organizations usually face a fundamental tradeoff between privacy and utility. They can choose not to publish any release candidate in order to keep their data perfectly private.[1] Or, they can choose to release the data without any modification to maximize data utility and provide no privacy protection. In this section, we present algorithms that make good tradeoffs between the two extremes. Intuitively, each algorithm tries to find the release candidate of a sanitization mechanism that satisfies a privacy criterion and maximizes a utility metric. We first discuss algorithms based on deterministic sanitization mechanisms (including suppression, generalization, microaggregation, bucketization, and decomposition), and then describe algorithms based on randomized sanitization mechanisms (including local randomization, input randomization, and synthetic data generation).

[1] Even the "perfect privacy" criterion cannot keep data perfectly private because the data publisher may fail to recognize a secret or sensitive query.

## 4.1 Deterministic Sanitization Techniques

Data recoding, generalization, suppression, and aggregation techniques have been used for many years to provide support for identity and privacy protection. Such techniques have the advantage of producing results that are semantically consistent with the input (sometimes described as "truthful" output), and they have been used to implement a variety of privacy requirements, including $k$-anonymity, $\ell$-diversity, and variations incorporating more complex models of adversarial background knowledge.

In this section, we describe techniques and algorithms for data suppression, microdata recoding, structured aggregation, microaggregation, and decomposition. These approaches share a number of clear similarities, but there are also some subtle differences.

In the interest of clarity, we will use the term *microdata* to refer to a non-aggregate data set (as found, for example, in a single relational database table). We will use the term *contingency table* to refer to a cross-tabulation of counts, as obtained, for example, using an SQL GROUP BY query.

### 4.1.1 Suppression-Based Mechanisms

One of the simplest techniques that can be used to implement privacy requirements such as $k$-anonymity is suppression of selected cells in the input microdata set $D$. For example, the release candidate shown in Table 4.1(b) replaces certain cells from the microdata in Table 4.1(a)

Table 4.1. Example of cell-suppression anonymization.

(a) Original table

| Zip code | Gender | Disease |
|---|---|---|
| 94085 | M | HIV |
| 14085 | M | HIV |
| 14085 | F | None |
| 94085 | F | HIV |
| 14085 | F | Flu |
| 14085 | F | None |
| 14085 | F | None |
| 14085 | F | Flu |

(b) Anonymized table

| Zip code | Gender | Disease |
|---|---|---|
| * | * | HIV |
| * | * | HIV |
| 14085 | F | None |
| * | * | HIV |
| 14085 | F | Flu |
| 14085 | F | None |
| 14085 | F | None |
| 14085 | F | Flu |

with wildcard values, denoted "*". In this case, for QI attributes *Zip code* and *Gender*, this suppression is sufficient to obtain 3-anonymity.

If we view the number of cells suppressed from $D$ as a rough indicator of data utility, then the problem of optimal $k$-anonymization is easily formulated in terms of the suppression function $s()$ producing $k$-anonymous output $D^*$ that suppresses the fewest cells.

This simple version of the problem has been widely studied. Meyerson and Williams [178] and Aggarwal et al. [13] both proved that the problem is NP-hard. Meyerson and Williams provide an $O(k \log k)$ approximation algorithm [178], meaning that the number of cells suppressed by their algorithm is guaranteed to be within a factor of $O(k \log k)$ of the optimal. Aggarwal et al. improve this result to obtain an $O(k)$ approximation [13], and Park and Shim further refined the result to obtain an $O(\log k)$ approximation [200].

### 4.1.2 Generalization-Based Mechanisms

Rather than making a binary decision for each data value (i.e., to suppress the data value, or preserve it in its original form), intuition says that we should be able to obtain better data utility by allowing for the *generalization* of certain values through a number of intermediate states.

In the input microdata $D$, there is a domain (e.g., dates, five-digit integers, etc.) associated with each attribute. We denote the domain of attribute $A$ as $dom(A)$. Based on this original input domain, it is possible to construct a more "general" and semantically consistent domain in a variety of ways. For example, the domain of attribute City can be generalized by replacing city values with states, and integer values can be replaced with ranges.

For categorical attributes, this idea of generalization can be implemented through the user-defined *generalization hierarchies* proposed by Samarati and Sweeney [226, 241]. Formally, such a hierarchy is defined by a set of many-to-one value generalization functions. Each generalization function $\gamma : dom(A) \rightarrow dom(A')$ maps each value in $dom(A)$ to a semantically consistent value in domain $dom(A')$. For example, Figure 4.1 shows a value generalization hierarchy for the *Nationality*

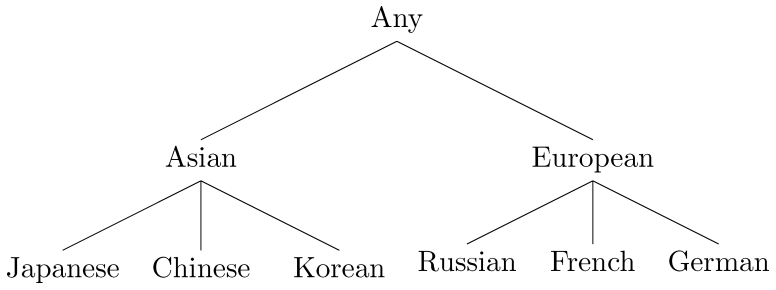



Fig. 4.1 Example generalization hierarchy for nationality.

attribute. Values of $dom(Nationality)$ are shown at the leaves of the tree. Notice that $\gamma(Japanese) = Asian$ and $\gamma(Asian) = Any$.

We will refer to the *height* of the generalization hierarchy as the total number of generalizations that can be applied. (E.g., The height of the hierarchy in Figure 4.1 is 2.) We will use the notation $\gamma^+(a)$ to refer to the generalization closure for input value $a$. For example, $\gamma^+(Japanese) = \{Japanese, Asian, All\}$.

While user-defined generalization hierarchies are well-suited to unordered categorical attributes, numeric attributes permit an additional degree of flexibility. In this case, generalization can instead take the form of a coarsened range of values. For example, we might replace the age value 22 with the range [18–24]. Alternatively, we could choose a different, yet still consistent, range [22–28]. This is also closely related to the classical ideas of *top-coding* and *bottom-coding*, commonly used in official statistics. For example, we might replace the age value 99 with the top-coded range [90–$\infty$].

#### 4.1.2.1 Local Recoding

Incorporating user-specified generalization hierarchies, it is possible to generalize the basic cell-suppression problem described in Section 4.1.1. We will refer to this new problem as *local recoding*, where each record can be generalized at a different granularity from the other records, even when they have the same attribute values.

In this case, it is easy to think of the generalization process as applying the generalization functions $\gamma()$ (often repeatedly) to the

individual cells of input microdata $D$. One way to quantify the utility of the resulting data is to count invocations of the $\gamma()$ function. In this case, the optimal $k$-anonymous generalization problem can be stated in terms of the generalization function $g()$ such that $D^* = g(D)$ satisfies $k$-anonymity and $g()$ minimizes invocations of $\gamma()$. Of course, minimum cell suppression is a special case of this more general problem; thus, the local recoding problem is also NP-hard. Sweeney's DataFly system provides some simple heuristic algorithms for this problem [239, 241, 238], and Aggarwal et al. provide an $O(k)$ approximation algorithm for the local recoding problem as well [13].

#### 4.1.2.2 Global Recoding and Structured Aggregation

Another class of techniques seeks to recode or generalize the domain of each QI attribute in the input microdata $D$. This can be done by treating each attribute independently (called single-dimensional recoding), or by recoding the domain of $n$-vectors (called multidimensional recoding). Note that the domain of $n$-vectors is the cross product of the domains of individual attributes.

---

**Definition 4.1 (Single-Dimensional Global Recoding).** A single-dimensional global recoding for input data set $D$ with QI attributes $Q_1, \ldots, Q_n$ is defined by a family of $n$ generalization functions $\phi_i : dom(Q_i) \rightarrow dom(Q_i')$, such that the values in $dom(Q_i')$ are semantically consistent generalizations of the values in $dom(Q_i)$.

---

For example, consider the input data in Table 4.2, with QI attributes *Nationality* and *Age*. A 2-anonymous single-dimensional recoding is shown in Table 4.3(a). Notice also that, for each single-dimensional global recoding, there exists a corresponding (partially aggregated) contingency table, expressed over the QI attributes, as shown in Table 4.3(b). Note that, unlike local recoding, if two tuples share the same value of an attribute, then after single-dimensional global recoding, the two tuples will share the same generalized value of that attribute.

The multidimensional recoding approach loosens the restrictions on generalization functions.

Table 4.2. Example input data.

| Nationality | Age | Disease |
|---|---|---|
| Russian | 20 | HIV |
| Russian | 21 | Flu |
| Russian | 24 | Flu |
| German | 21 | Cancer |
| German | 23 | Hepatitis |
| French | 25 | Cancer |
| French | 25 | HIV |
| Japanese | 20 | Bronchitis |
| Japanese | 24 | Hepatitis |
| Japanese | 25 | Flu |
| Chinese | 22 | Hepatitis |
| Korean | 21 | Flu |

Table 4.3. Single-dimensional global recoding.

(a) Expressed as microdata

| Nationality | Age | Disease |
|---|---|---|
| European | [20–22] | HIV |
| European | [20–22] | Flu |
| European | [23–25] | Flu |
| European | [20–22] | Cancer |
| European | [23–25] | Hepatitis |
| European | [23–25] | Cancer |
| European | [23–25] | HIV |
| Asian | [20–22] | Bronchitis |
| Asian | [23–25] | Hepatitis |
| Asian | [23–25] | Flu |
| Asian | [20–22] | Hepatitis |
| Asian | [20–22] | Flu |

(b) Expressed as a 2D aggregate contingency table

| | European | | | Asian | | |
|---|---|---|---|---|---|---|
| | French | German | Russian | Japanese | Chinese | Korean |
| [20–22] | | 3 | | | 3 | |
| [23–25] | | 4 | | | 2 | |

**Definition 4.2 (Multidimensional Global Recoding).** A multidimensional global recoding for input data set $D$ with QI attributes $Q_1, \ldots, Q_n$ is defined by a single generalization function $\phi : dom(Q_1) \times \cdots \times dom(Q_n) \rightarrow dom(Q')$, where the values in $dom(Q')$ are $n$-dimensional vectors that are semantically consistent with values in $dom(Q_1) \times \cdots \times dom(Q_n)$.

For example, consider again the input data from Table 4.2. A multidimensional global recoding is shown in Table 4.4. Notice that, under this less-restrictive variation, we can further refine the *Age* values of Europeans beyond what was permitted under single-dimensional recoding (Table 4.3), while still satisfying 2-anonymity.

A variety of algorithms have been proposed for single-dimensional and multidimensional global recodings, including optimal search over various restricted spaces of generalizations [29, 149, 226], randomized search [132, 267], and heuristic search [107, 150, 151, 261]. Many of these algorithms can be applied to enforce more than one privacy requirement (e.g., $k$-anonymity, $\ell$-diversity, etc.). For example, the recoded data in Table 4.4(a) satisfy both 2-anonymity and entropy 2-diversity.

To give a concrete example, one such algorithm (*Mondrian*) is based on greedy recursive spatial partitioning, and is shown in Algorithm 1. As input, this algorithm takes the $d$-dimensional quasi-identifier domain space (i.e., $dom(Q_1) \times \cdots \times dom(Q_n)$), which is

Table 4.4. Multidimensional global recoding.

(a) Expressed as microdata

| Nationality | Age | Disease |
|---|---|---|
| European | [20–22] | HIV |
| European | [20–22] | Flu |
| European | [23–24] | Flu |
| European | [20–22] | Cancer |
| European | [23–24] | Hepatitis |
| European | 25 | Cancer |
| European | 25 | HIV |
| Asian | [20–22] | Bronchitis |
| Asian | [23–25] | Hepatitis |
| Asian | [23–25] | Flu |
| Asian | [20–22] | Hepatitis |
| Asian | [20–22] | Flu |

(b) Expressed as a 2D aggregate contingency table

| | European | | | Asian | | |
|---|---|---|---|---|---|---|
| | French | German | Russian | Japanese | Chinese | Korean |
| [20–22] | | 3 | | | 3 | |
| [23–24] | | 2 | | | 2 | |
| 25 | | 2 | | | | |

denoted $G$, a data set $D$, and a privacy requirement $\rho$. The algorithm greedily partitions the domain space, at each step choosing the axis-parallel split that optimizes some objective function without violating privacy requirement $\rho$. (In the pseudo-code, the functions $ChooseAttr()$ and $ChooseThresh()$ denote the selection of a dimension upon which to split, as well as the threshold point about which to split. Resulting regions are captured using ranges $max - min$ for numeric attributes, and values in the generalization hierarchy for categorical attributes.) In the case of nominal (unordered categorical) attributes, these splits are further restricted by user-provided value generalization hierarchies.

The correctness of Mondrian (and many of the other algorithms) relies on two properties of the privacy requirement $\rho$: monotonicity and bucket independence, which are defined below. It is easy to think of the recoding function $\phi$ as partitioning the input data set $D$ into a set of non-overlapping buckets, each with identical quasi-identifier values. In the following, we define a partial order on the set of all partitionings of input data $D$. We say that partitioning $D_1^* \preceq D_2^*$ if and only if each bucket in $D_2^*$ is the union of one or more buckets in $D_1^*$. In the following, the notation $\rho(D^*)$ indicates that $D^*$ satisfies privacy requirement $\rho$.

---

**Definition 4.3 (Monotonicity Property).** Let $D_1^*$ and $D_2^*$ be partitionings of input data $D$ such that $D_1^* \preceq D_2^*$. A privacy requirement $\rho$ satisfies the monotonicity property iff $\rho(D_1^*) \rightarrow \rho(D_2^*)$.

---

**Definition 4.4 (Bucket Independence Property).** Let $D_1$ and $D_2$ be disjoint tuple sets, and let $D_1^*$ and $D_2^*$ be partitionings of $D_1$ and $D_2$, respectively. A privacy requirement $\rho$ satisfies bucket independence iff $\rho(D_1^*) \wedge \rho(D_2^*) \rightarrow \rho(D_1^* \cup D_2^*)$.

---

The Mondrian algorithm can be used to implement privacy requirements $\rho$ satisfying the monotonicity and bucket independence properties. Examples of such requirements include $k$-anonymity, entropy $\ell$-diversity, and recursive $(c, \ell)$-diversity. The bucket independence property allows for the recursive decomposition of the problem, while the monotonicity property guarantees that the output is *minimal*

**Algorithm 1** Mondrian domain-space partitioning

**Input:** QI domain space $G$, data set $D$, privacy requirement $\rho$
**Output:** recoding function $\phi$

```
1: if no allowable split for G, D under ρ then
2:   return φ : t ∈ D → tuple representation of G, D
3: else
4:   best ←ChooseAttr(D, {Q_1,...,Q_d}, ρ)
5:   if numeric(best) or ordinal(best) then
6:     threshold ← ChooseThresh(best, D, ρ)
7:     D_1 ← {t : t ∈ D, t.best ≤ threshold}
8:     D_2 ← {t : t ∈ D, t.best > threshold}
9:     G_1 ← Update G by setting best.max = threshold
10:    G_2 ← Update G by setting best.min = threshold
11:    return Mondrian(G_1,D_1,ρ) ∪ Mondrian(G_2,D_2,ρ)
12:  else if nominal(best) then
13:    recodings ← {}
14:    for all child v_i of root(best.hierarchy) do
15:      D_i ← {t : t ∈ D, t.best descended from v_i in best.hierarchy}
16:      G_i ← Update G by setting best.value = v_i
17:    end for
18:    Q' ← Replace best.hierarchy with subtree rooted at v_i in
       {Q_1,...,Q_d}
19:    recodings ← recodings ∪ Mondrian(G_i, D_i, Q', ρ)
20:    return recodings
21:  end if
22: end if
```

(i.e., no partition can be further divided without violating the privacy requirement). It is important to note that the algorithm does not directly apply to privacy requirements that do not satisfy bucket independence, including $(c,k)$-safety (Definition 2.3) and the 3D privacy criterion (Definition 2.4). However, Chen et al. adapted the algorithm to the 3D criterion by incorporating a set of constant-sized global summary statistics [51]. Later work considered scaling variations of this algorithm to large data sets [131, 152].

### 4.1.3 Microaggregation

Another related technique for releasing microdata is called *microaggregation*. Microaggregation conceptually involves two different phases: *data partitioning* and *partition aggregation* [79]. During the first phase, the input microdata $D$ is partitioned into subsets $D_1, \ldots, D_n$ such that $D_i \cap D_j = \emptyset$ (for $i \neq j$) and $D_1 \cup \cdots \cup D_n = D$.[2] This involves [69] (a) segmenting the set of attributes into $s$ parts, where each segment contains similar attributes, (b) partitioning each segment into groups that satisfy a privacy constraint (like $k$-anonymity) and a homogeneity constraint (e.g., minimizing a norm between the largest and smallest elements) for utility. A variety of such partitioning and clustering algorithms have been proposed [14, 69, 79, 253].

Following the partitioning phase, the data in each partition $D_i$ are replaced with one or more aggregate values (e.g., sum, variance, median, etc.). For example, consider the input data shown in Table 4.5(a) and a possible microaggregated version of this data set shown in Table 4.5(b).

Alternatively, rather than replacing each cluster with one or more aggregates, Aggarwal and Yu [10] proposed generating synthetic data based on the aggregate properties of each partition.

Table 4.5. Example of microaggregation.

| (a) Original Data | | (b) Aggregated data | |
|---|---|---|---|
| Age | Sex | Mean(Age) | Mode(Sex) |
| 25 | M | 22.67 | M |
| 23 | M | 22.67 | M |
| 20 | F | 22.67 | M |
| 27 | F | 23.33 | F |
| 19 | F | 23.33 | F |
| 24 | F | 23.33 | F |
| 40 | F | 35 | F |
| 30 | F | 35 | F |

[2] This phase is subtly different from the partition phase of global recoding algorithms such as Mondrian [150], which are based on partitioning the domain space, rather than the data.

Table 4.6. Example bucketized data.

| Bucket | Nationality | Age |
|---|---|---|
| 1 | Russian | 20 |
| 1 | Russian | 21 |
| 1 | Russian | 24 |
| 2 | German | 21 |
| 2 | German | 23 |
| 2 | French | 25 |
| 2 | French | 25 |
| 3 | Japanese | 20 |
| 3 | Japanese | 24 |
| 3 | Japanese | 25 |
| 4 | Chinese | 22 |
| 4 | Korean | 21 |

| Bucket | Disease |
|---|---|
| 1 | HIV |
| 1 | Flu |
| 1 | Flu |
| 2 | Cancer |
| 2 | Hepatitis |
| 2 | Cancer |
| 2 | HIV |
| 3 | Bronchitis |
| 3 | Hepatitis |
| 3 | Flu |
| 4 | Hepatitis |
| 4 | Flu |

### 4.1.4 Bucketization

Also building on similar intuition, recent work has considered using a bucketization technique to achieve $\ell$-diversity [273]. Like microaggregation, bucketization partitions the input data $D$ into non-overlapping "buckets." However, rather than summarizing each bucket, the bucketization approach simply breaks the connection between quasi-identifier and sensitive attributes. For example, Table 4.6 shows a bucketized representation of the input data from Table 4.2. The drawback of this approach is that its application is limited to privacy definitions based on clearly defined sensitive attributes (e.g., $\ell$-diversity). Also, since the quasi-identifier attributes are released without any modification, an adversary is likely to be able to identify the records of some individuals by a link attack. Although the predefined sensitive attribute values of those individuals are not identified, allowing an adversary to pinpoint your record in a published data set is sometimes considered to be undesirable. Also, once a link is established it may be possible to re-establish a probabilistic relationship between a tuple and its sensitive value by building a statistical model over the sanitized data (see Section 6.1.3).

### 4.1.5 Decomposition and Marginals

Finally, a significant amount of work has focused on marginalization and decomposition techniques for identity and attribute protection

[75, 77, 78, 142, 235]. The idea of marginalization is, given a full joint contingency table as input, "sum out" selected subsets of the attributes to produce lower-dimensional (marginal) contingency tables (these can be thought of as histograms on subsets of the attributes). Similarly, in decomposition, the idea is to take a single table of microdata as input, and project on selected attribute subsets. Past work is this area has considered computing upper and lower bounds on cell counts in the original contingency table, given marginals [75, 78, 235], as well as techniques for probabilistically reasoning about disclosure of a sensitive attribute [75, 142].

## 4.2 Randomized Sanitization Techniques

Generalization, aggregation, and suppression are attractive privacy mechanisms since they only output truthful facts about the original data. However, data collected by most organizations like the Census Bureau are incomplete, imprecise, and sometimes uncertain. Moreover, aggregation techniques do not satisfy very strict privacy criteria like differential privacy. This led to the development of privacy mechanisms based on random perturbations of the original data such that the perturbed data retain the statistical properties of the input data. As we will see in this section, these techniques are reasonable since the input data themselves are an approximation of the truth and, in addition, these techniques are able to guarantee stronger privacy than aggregation techniques. We describe the following randomization techniques for privacy — *local randomization*, *input randomization*, *perturbing statistics*, *statistics-preserving input randomization*, and *model-based synthetic data generation*.

### 4.2.1 Local Randomization Techniques

Data collectors including the Census Bureau do not obtain truthful answers to all questions on their surveys. Respondents do not trust the data collectors especially when answering sensitive questions (e.g., *"Have you ever used illegal drugs?"*). Local randomization techniques have been used to elicit truthful answers. As the name suggests, in these techniques each individual respondent randomly perturbs his/her

data before sharing it with the the data collector. The randomization operators are designed such that (a) they preserve the privacy of the individuals, while (b) allowing the data collector to learn statistical properties of the data if a sufficiently large amount of data is collected.

Warner's *randomized response* technique [262] for answering sensitive Boolean questions is the earliest local randomization technique. Here each individual $i$ independently answers a *yes/no* question $Q$ as follows: $i$ answers truthfully with probability $p_i$, and lies with probability $(1 - p_i)$. Given $n$ such perturbed answers, the aggregate answer can be estimated as follows (when all the respondents use the same probability $p$). Let $\pi$ be the fraction of the population for which the true response to $Q$ is *yes*. Then the expected proportion of *yes* responses is

$$P(yes) = \pi \cdot p + (1 - \pi) \cdot (1 - p) \tag{4.1}$$

$$\text{Hence, } \pi = \frac{P(yes) - (1 - p)}{2p - 1} \tag{4.2}$$

If $m$ out of the $n$ individuals answered *yes*, then the following $\hat{\pi}$ is an estimator for $\pi$.

$$\hat{\pi} = \frac{\frac{m}{n} - (1 - p)}{2p - 1} \tag{4.3}$$

Instead of lying with probability $(1 - p)$, respondents could also perturb their answers using a second scheme proposed by Warner. An individual answers the question posed by the data collector $(Q)$ honestly with probability $p$ and answers a different innocuous question $(Q_I)$ with probability $(1 - p)$. For instance, with probability $p$, the respondent truthfully answers if he/she had used illegal drugs, and with probability $(1 - p)$, the respondent flips a coin with bias $\alpha$ and answers *yes* if the respondent got a head. In this case, the probability that the answer to $Q_I$ is *yes* is $\alpha$. If $m$ out of the $n$ individuals answered *yes*, an estimator for $\pi$ is derived below.

$$P(yes) = \pi \cdot p + \alpha \cdot (1 - p) \tag{4.4}$$

$$\pi = \frac{P(yes) - (1 - p) \cdot \alpha}{p} \tag{4.5}$$

$$\bar{\pi} = \frac{\frac{m}{n} - (1 - p) \cdot \alpha}{p} \tag{4.6}$$

$\bar{\pi}$ has a smaller variance than $\hat{\pi}$ when the probability of answering the correct question $p$ is not too small. Hence, typically the innocuous question technique is better than naive randomized response.

Randomized response techniques can be proven to guarantee $(\alpha, \beta)$-privacy using the $\gamma$ amplification condition (Section 2.5). Warner's original technique has an amplification of

$$\max\left(\frac{p}{1-p}, \frac{1-p}{p}\right)$$

To maximize utility, the probability of lying should be smaller ($p > 1 - p$). Moreover, $(\alpha, \beta)$-privacy is guaranteed if $p$ satisfies the following condition,

$$\frac{p}{1-p} < \frac{\beta}{\alpha} \cdot \frac{1-\alpha}{1-\beta}$$

$$\text{or if, } p < \frac{\beta(1-\alpha)}{\beta(1-\alpha) + \alpha(1-\beta)}$$

Subsequent work [92, 18] generalized the above randomized response techniques to other domains. Each record $u \in U$ corresponds to the sensitive information of a distinct individual. Each $u$ is independently randomized using a perturbation matrix $\mathcal{A}$; the entry $\mathcal{A}[u, v]$ describes the transition probability $P(u \to v)$ of perturbing a record $u \in D_U$ to a value $v$ in the perturbed domain $D_V$. The matrix $\mathcal{A}$ should satisfy the following properties:

$$\mathcal{A} \geq 0, \quad \sum_{v \in D_V} \mathcal{A}[u, v] = 1 \quad \forall u \in D_U \tag{4.7}$$

Evfimievski et al. [92] studied the problem where individuals share itemsets (e.g., set of movies rented) with an untrusted server (e.g., an online movie rental company) in return for services (e.g., movie recommendations), and were the first to propose a formal definition of privacy breaches using the $(\rho_1, \rho_2)$-privacy definition. Here, the purpose of collecting itemsets is to identify those sets of items that occur frequently across users (for example, movies that tend to be rented together). Evfimievski et al. showed that itemsets randomized using Algorithm 2, with parameters $\rho$ and $\{p[j]\}_{j=0}^{m}$, both preserve privacy

**Algorithm 2** Select-a-Size Algorithm

**Input:** Itemset $\mathcal{I}_u \subseteq D_U$, $|\mathcal{I}_u| = m$.

**Output:** Randomized itemset $\mathcal{I}'_u$.

1: Select an integer $j \in [1, m]$, with probability $p[j]$.
2: $\mathcal{I}'_u \leftarrow$ simple random sample of size $j$ of $\mathcal{I}_u$.
3: For every $a \in D_U \setminus \mathcal{I}_u$, add $a$ to $\mathcal{I}'_u$ with probability $\rho$.

and allow a data collector to correctly estimate the frequent itemsets. Later, Agrawal and Haritsa [18] improved on this by finding an optimal perturbation matrix $\mathcal{A}$.

Evfimievski et al. [92] proved sufficient conditions on the parameters, $\rho$ and $\{p[j]\}_{j=0}^{m}$, in order to satisfy the $\gamma$-amplification condition and simultaneously maximize the utility of the randomization method (e.g., maximizing $|I_u \cap I'_u|$, the number of original items retained in the randomized itemset). Algorithms for recovering the original data from the randomized itemsets and for producing unbiased estimators for the mean and the covariance of these estimates are provided in [93].

### 4.2.2 Input Randomization Techniques

While local randomization techniques protect the privacy of individuals right at the stage of data collection, there are many scenarios where fairly accurate data are being collected from individuals. Examples include search queries collected by search engine companies and movie ratings collected by companies like Netflix. These organizations would like to extract user statistics from these collected data $D$ without disclosing personal information. One way to achieve this could be to apply a local randomization technique on $D$; i.e., independently perturb the records of each individual in $D$ to get $D'_{ind}$ and use $D'_{ind}$. However, since the data collector has access to the complete data $D$ in this case, more interesting randomization operators could be used to perturb groups of records. Intuitively, such a methodology should provide strictly more utility since we are allowed to use a richer set of perturbation schemes. We call the class of such methods as *input randomization* techniques.

**Additive Perturbation** A simple technique to perturb numeric data, proposed by Agrawal and Srikant [16], is to independently add 0-mean noise to each record. Let $V$ be a noise matrix, then the perturbed data are $U_p = U + V$. The random noise added to each record ($v \in V$) is usually either a uniform random variable in $[-\alpha, \alpha]$ or distributed as a Gaussian with 0 mean and a known variance. The privacy of such a scheme is unclear; in fact, if the random noise variables are uncorrelated, Kargupta et al. [138] and Huang et al. [126] showed that very accurate estimates of the original data can be recovered from such additively perturbed data due to dependencies inherent in $U$. For instance, suppose an adversary knows that all the records in $U$ have the same value, say $z$. Then, additive randomization does not guarantee any privacy; the mean of the perturbed data accurately estimates $z$ if there are enough records in $U$.

Additive randomization can be broken using Principal Component Analysis (PCA). Suppose the data have $m$ dimensions and are perturbed by adding noise independently to each dimension. Usually, different attributes in the data are correlated; hence, the data can be projected onto a smaller number, $p < m$, of dimensions. The first principal component (PC) of the data is the direction, $e_1$, along which the data have the highest variance. The $i$-th PC, $e_i$, is a vector orthogonal to the first $(i-1)$ PCs with the largest variance. These vectors are the eigenvectors of the covariance matrix of the data. In correlated data, only the variances along $p$ directions are large. However, for the random noise, the variances are the same along all directions. The variances of the perturbed data are roughly the sum of the variances of the original data and the random noise. Hence, by dropping $(m-p)$ directions along which the perturbed data have the least variance, while much information is not lost about the original data, a $(1 - p/m)$ fraction of the noise added is removed; this might lead to privacy breaches.

**Post Randomization** The post-randomization method (PRAM) [117] is very similar in spirit to local randomization techniques. Suppose every entry in the database takes values in $1 \ldots K$. Then PRAM randomly perturbs each entry in the database to some other value in $[1, K]$. Let $p_{k\ell}$ denote the probability that value $k \in [1, K]$ is transformed to

a value $\ell$. Let $P = \{p_{k\ell}\}$ be a Markov $K \times K$ matrix with $p_{k\ell}$ as its $(k,l)$-th entry. It is easy to see that the privacy guaranteed by PRAM is identical to that of local randomization. Moreover, the original data can be regenerated from the perturbed data using the following unbiased estimator:

$$\hat{T} = (P^{-1})^t T^\star,$$

where $T = \{T_1, \ldots, T_k\}^t$ is the vector of counts, such that $T_i$ is the number of entries in the database with value $i$, and $T^\star$ is a similar vector of counts on the perturbed database.

Since PRAM perturbs the data after collection, one could potentially choose $P$ based on the data distribution. For instance, in *invariant PRAM*, $P$ is chosen such that $\|P^t T - T\| < \epsilon$, in particular $\epsilon$ can be 0. The advantage of such a perturbation matrix is that the perturbed vector of counts is itself an unbiased estimator of $T$. That is the perturbed database can be used directly instead of the original database (without multiplying by $P^{-1}$).

Note that $P = I$, the identity matrix, always satisfies this constraint (equivalent to publishing the original data), but is uninteresting. A nontrivial $P$ can be constructed as follows. Let $m \in [1, K]$ be the category appearing the smallest number of times in the database; i.e., $T_m$ is the smallest count. Then, for some $0 \leq \theta \leq 1$, let

$$p_{kl} = \begin{cases} 1 - \theta \cdot \frac{T_m}{T(k)}, & k = \ell; \\ \frac{\theta}{K-1} \cdot \frac{T_m}{T(k)}, & k \neq \ell. \end{cases}$$

However, as $K$ becomes very large, $T_m$ tends to 0. Hence, it becomes very hard to find an invariant $P$. Hence, the authors [117] suggest that it is probably best to apply PRAM such that some distributions/marginals are preserved while others are not.

### 4.2.3 Perturbing Statistics

Sometimes the desired sanitized data are just a set of statistics that describe the original data (for example, the mean, median, etc.). In these cases, there is a strong similarity between privacy-preserving query answering (in statistical databases) [4] and privacy preserving

data publishing. In this section we will review some of the most important ideas from query answering that are applicable to data publishing. These include negative results on which statistics can be published, and positive results on how to publish them.

Adding noise to every tuple in the database may not guarantee privacy but may still cause a large distortion in the data. Recent work has shown that it is possible to add noise directly to statistics of interest while guaranteeing strong privacy [74, 37, 87, 192]. The general framework is (a) list out the statistics of interest $Q_1, \ldots, Q_k$, (b) independently draw $k$ samples $\eta_1, \ldots, \eta_k$ from a preferably heavy-tailed distribution (such as the Laplace distribution, although a Gaussian distribution $N(0, \sigma^2)$ is sometimes used), and (c) return the noisy statistics $Q_1(D) + \eta_1, \ldots, Q_k(D) + \eta_k$. The statistics of interest could be the complete contingency table (where $\{1, \ldots, k\}$ represent the domain from which all the values in the database are drawn, and each $Q_i$ is the count of records in the database with value $i$), or a set of marginals, or an arbitrarily complex set of aggregate queries (like in statistical databases). Hence, we will use the term query instead of statistic in the rest of this section. The key contribution of this line of work is that adding noise proportional to the *sensitivity* of a query guarantees differential privacy. We next describe how to add Laplacian noise to achieve differential privacy, then describe an extension of this technique to publish marginals of a contingency table, and conclude with limits to this approach.

**Laplacian Noise Addition and Query Sensitivity.** Let $Q : dom(D) \to \mathcal{R}$ be a statistic. Define the *sensitivity* of query $Q$ to be the smallest number $S(Q)$, such that

$$\forall U_1, U_2 \text{ that differ in one record, } |Q(U_1) - Q(U_2)| \leq S(Q) \tag{4.8}$$

Let $Lap(\lambda)$ denote the *Laplace* distribution which has a density function $h(y) \propto \exp(-|y|/\lambda)$. Suppose a query $Q(U)$ posed to a database $U$ is answered using $Q(U) + Y$, where $Y \sim Lap(S(Q)/\epsilon)$. This perturbation scheme indeed satisfies $\epsilon$-differential privacy. For every $U_1, U_2$ that differ in only one record $u_i$,

$$\frac{P(Q(U_1) + Y = x)}{P(Q(U_2) + Y = x)} = \frac{h(x - Q(U_1))}{h(x - Q(U_2))} \tag{4.9}$$

$$= \frac{\exp(-|x - Q(U_1)| \times \epsilon/S(Q))}{\exp(-|x - Q(U_2)| \times \epsilon/S(Q))} \tag{4.10}$$

$$\leq \exp(\epsilon \times |Q(U_1) - Q(U_2)|/S(Q)) \tag{4.11}$$

$$\leq \exp(\epsilon) \tag{4.12}$$

For instance, consider publishing the entire contingency table of counts (also called histogram query). The sensitivity of the histogram query is 2 — changing the value of one individual from $i$ to $j$, reduces the count $Q_i$ by 1, and increases the count $Q_j$ by 1. Hence, one can publish the entire contingency table by adding noise drawn from $Lap(2/\epsilon)$. There are still a couple of problems — (a) we can get fractional or negative counts and (b) for a sparse domain where most values have count 0, the total noise added might be very large. We will describe Barak et al.'s [27] techniques to handle (a).

Barak et al. [27] propose a solution to publish marginals of a contingency table using the Laplacian noise-addition. Publishing a set of noise-infused marginals is not satisfactory; such marginals may not be consistent, i.e., there may not exist a contingency table that could simultaneously generate all these marginals, and the resulting "counts" in the the noise-infused marginals may even be negative. Barak et al. solve this problem by adding noise to a small number of Fourier coefficients; any set of Fourier coefficients correspond to a (fractional and possibly negative) contingency table. They show that only a "small" number of Fourier coefficients are required to generate the required marginals, and hence only a small amount of noise (proportional to the size of the marginal domain) is required. In order to create non-negative and integral marginals, the authors employ a linear program solution (in time polynomial in the size of multidimensional domain) to generate the final non-negative integral set of noise-infused marginals.

**Instance Specific Noise.** Nissim et al. [192] proposed a novel algorithm where the noise added to a statistic is not only calibrated to the sensitivity of the query, but also to the input database. The new scheme, which allows for much less noise to be added in many cases, has the following intuition. Suppose $U$ is the input data. Let $LS(Q, U)$

be the maximum value of $|Q(U) - Q(U')|$ over all $U'$ that differ from $U$ in one tuple. Nissim et al. term $LS(Q,U)$ as the *local sensitivity* of $Q$ at $U$. Let us consider the impact of adding $Lap(LS(Q,U)/\epsilon)$ on both utility and privacy.

It is easy to see that $S(Q)$ is the maximum local sensitivity of $Q$ over all inputs $U$. Hence, the local sensitivity of a function may be much smaller than the (global) sensitivity of a query on some inputs. For instance, let the database have five tuples, and suppose each tuple in the database takes values between 0 and $\Lambda$. Let $Q$ be the function that returns the median of these tuples. The sensitivity of $Q$ is $\Lambda$. Let $U = [0,0,\Lambda,\Lambda,\Lambda]$ and $U' = [0,0,0,\Lambda,\Lambda]$; $Q(U) = 0$, while $Q(U') = \Lambda$. On the contrary, the local sensitivity of the function is only $\max\{x_2 - x_3, x_3 - x_4\}$, where $x_i$ is the $i$-th largest tuple in the database; this could be as small as 0. Hence, there will be many cases when $Lap(LS(Q,U)/\epsilon)$ will be much smaller than $Lap(S(Q)/\epsilon)$ and thus guarantee more utility.

On the privacy front, one may be wrongly led to believe that adding $Lap(LS(Q,U)/\epsilon)$ noise satisfies differential privacy; after all, Equations (4.9)–(4.12) hold even if $S(Q)$ is replaced with $LS(Q,U_1)$. Unfortunately, privacy is not guaranteed since the *amount* of noise added also leaks information to the adversary; especially when $LS(Q,U_1)$ is much different from $LS(Q,U_2)$ for $U_1$ and $U_2$ differing in one tuple. For instance, when the local sensitivity is 0 (like in the previous example), no noise will be added, and thus one cannot expect privacy. Nissim et al. instead proposed adding noise proportional to a *smooth upper bound* of $LS(Q,U)$, and showed that this guarantees differential privacy.

---

**Definition 4.5 (Smooth Upper Bound).** A function $\mathcal{S}(Q,U)$ is defined to be a $\beta$-smooth upper bound of $LS(Q,U)$ if

- $\forall U$, $\mathcal{S}(Q,U) \geq LS(Q,U)$.
- $\forall U_1, U_2$, that differ in one tuple, $\mathcal{S}(Q,U_1) \leq \beta \mathcal{S}(Q,U_2)$.

---

Note that $S(Q) = \max_U LS(Q,U)$ is one such smooth upper bound. Nissim et al. [192] proposed techniques for computing smooth upper bounds and provided noise distributions that could be used to protect privacy.

**Impossibility results** There are inherent limits to the amount information that can be published by adding noise to statistics. For instance, if multiple copies of the same statistic are published $m$ times with answers $X_1, \ldots, X_m$, where $X_i = Q(D) + \epsilon_i$, $\epsilon_i \sim N(0, \sigma^2)$, then

$$\epsilon = \frac{\sum_i \epsilon_i}{m} \sim N(0, \sigma^2/m)$$

Therefore, for a sufficiently large $m$, the average of all the $X_i$s is a very good estimator of the real answer $Q(D)$. Motivated by this intuition, Dinur and Nissim [74] showed the following stronger results.

Consider a database with $n$ tuples, where each tuple is a bit, i.e., $D \in \{0,1\}^n$. Consider an interactive algorithm[3] $\mathcal{A}$ that allows an adversary to pose *subset-sum* queries. Each subset-sum statistic $Q \subseteq [n]$ defines a subset of tuples in the database, and $Q(D)$ is the sum of these tuples.

$\mathcal{A}$ is defined to be within $\mathcal{E}$ *perturbation* if

$$\forall \text{ query } Q \in \mathcal{Q}, |\mathcal{A}(Q(D)) - Q(D)| < \mathcal{E}$$

$\mathcal{A}$ is said to be *non-private* if an adversary can efficiently reconstruct the entire database accurately.

---

**Definition 4.6 (Non-Privacy).** An $\mathcal{A}$ operating on database $D \in \{0,1\}^n$ is said to be $t(n)$-*non-private* if for every $\epsilon > 0$ there exists an adversary $\mathcal{M}$ who runs in time $t(n)$ and outputs a database $C$ such that:

$$Pr[\mathcal{M}^{\mathcal{A}}(1^n) \textit{ outputs } C \textit{ s.t. } dist(C, D) < \epsilon n] \geq 1 - neg(n),$$

where $\mathcal{M}^{\mathcal{A}}$ denotes the algorithm used by the adversary with access to the output perturbation mechanism, $n$ is the size of $D$, $dist(C, D)$ is the number of tuples $C$ and $D$ differ in, $neg(n) \in o(\frac{1}{p(n)})$ for any polynomial $p(n)$, and the probabilities are taken over the coin tosses of $\mathcal{M}$ and $\mathcal{A}$.

---

[3] While the Dinur and Nissim's, [74] results are in the context of the interactive setting where queries can be posed adaptively by the adversary, there are obvious connections to the data publishing scenario where a set of queries are answered by the data publisher up front.

**Algorithm 3** Exponential time adversary

**Input:** Access to output randomizer $\mathcal{A}$ with perturbation within $\mathcal{E}$.
**Output:** A database $C$.

1: [Query Phase]
2: For all $Q \subseteq [n]$, get perturbed answer $\mathcal{A}(Q(D))$.
3: [Weeding Phase]
4: **for all** $C \in \{0,1\}^n$ **do**
5: if for all $Q \subseteq [n]$, $|Q(C) - \mathcal{A}(Q(D))| \leq \mathcal{E}$, return $C$ and halt.
6: **end for**

Dinur et al. first show that for every $\mathcal{A}$ within $o(n)$ perturbation, an adversary can accurately reconstruct $D$ using Algorithm 3, which runs in time exponential in $n$. Since $\mathcal{A}$ is within $\mathcal{E}$ perturbation, the database $D$ will never be weeded out; therefore, $\mathcal{M}$ always halts. One can show that $dist(C, D) < 4\mathcal{E} = o(n)$. Suppose on the contrary $dist(C, D) > 4\mathcal{E}$. Let $Q_1$ be the query that sums the tuples that have the value 0 in $D$ and 1 in $C$. Let $Q_2$ be the query that sums the tuples that have the value 1 in $D$ and 0 in $C$. Clearly, $|Q_1| + |Q_2| = dist(C, D) > 4\mathcal{E}$. Without loss of generality, let $|Q_1| \geq 2\mathcal{E} + 1$. Then, $Q_1(D) = 0$, $Q_1(C) > 2\mathcal{E}$, and $\mathcal{A}(Q_1(D)) < \mathcal{E}$. However, $Q_1(C) - \mathcal{A}(Q_1(D)) > \mathcal{E}$ and hence $C$ will never be output by Algorithm 3.

The above attack algorithm can, in fact, be improved to accurately reconstruct the database using $t(n) = n(\log n)^2$ queries as long as $\mathcal{A}$ is within a perturbation of $o(\sqrt{n})$. In the improved attack, an adversary first asks $t(n)$ random queries $Q_1, \ldots, Q_{t(n)}$. In the weeding phase the adversary uses the following linear program with unknowns $c_1, \ldots, c_n$ that correspond to the entries in the reconstructed database $C$:

$$A(Q_\ell(D)) - \mathcal{E} \leq Q_\ell(C) \leq A(Q_\ell(D)) + \mathcal{E} \quad 1 \leq \ell \leq t(n)$$
$$0 \leq c_i \leq 1 \qquad\qquad 1 \leq i \leq n$$

The adversary rounds off each $c_i$ to 1 if it is greater than $\frac{1}{2}$. Dinur et al. show that the reconstructed $C$ differs from $D$ in only $o(n)$ positions.

More recently, Dwork et al. [88] showed a much stronger result that any privacy mechanism, interactive or noninteractive, providing reasonably accurate $(o\sqrt{n})$ answers to approximately 0.761 fraction

of randomly generated weighted subset-sum queries, and arbitrary answers on the remaining ≈0.239 fraction, is non-private.

On the positive side, Dinur et al. also proposed a technique that provably protects against a bounded adversary who is allowed to ask only $\mathcal{T}(n) \geq polylog(n)$ queries by using additive perturbation of the magnitude $\tilde{O}(\sqrt{\mathcal{T}(n)})$. Building on this result, Blum et al. [37] propose the SULQ framework that answers up to a sub-linear number of aggregate queries by adding Laplacian noise while guaranteeing differential privacy [85].

In summary, Laplacian noise addition guarantees strong privacy and has been shown to provide researchers with useful data. However, the problem of choosing the correct set of statistics for publication is still an open question. Publishing the entire contingency table, for instance, will lead to a lot of noise (especially in the portions of the domain with no data). Also, too many statistics cannot be published without arbitrary perturbation. Hence, we describe next a model-based approach to publishing data.

### 4.2.4 Statistics-Preserving Input Randomization

First proposed by Dalenius and Reiss [60], data swapping attempts to preserve privacy of individuals by swapping values for one attribute between different individuals. More precisely, consider a database with $n \geq t$ attributes. Let $\mathcal{A} = \{A_1, \ldots, A_t\}$ be a subset of attributes whose marginal distribution must be preserved in the published table. Two individuals $x$ and $x'$ are said to be $(t-1, A_i)$-equivalent if they agree on every attribute in $\mathcal{A}$ except for possibly $A_i$. The data swapping algorithm first forms all equivalence classes of $(t-1, A_1)$-equivalent records, and within each equivalence class performs a *primitive swap* — picks any two records and swaps the values for the attribute on which they differ. This is then repeated for $(t-1, A_2)$-equivalent records, etc. Dalenius and Reiss state that privacy is preserved if for every attribute of every record, there is at least one database with the same $t$-order marginal statistics that assigns a different value to that attribute.

Greenberg and Moore present extensions to the original data swapping algorithm for masking ordinal attributes. In an unpublished 1987

manuscript, Greenberg proposed a data swapping algorithm for ordinal attributes that works on one attribute at a time. First, the ordinal attribute is sorted and for each value of the attribute, one computes its rank (i.e., the number of values larger than it). Swapping can only be performed between records whose ranks are within a pre-specified threshold $\alpha$. This algorithm was shown to preserve statistics with acceptable error. Moore's algorithm [181] enhances this rank-based proximity swapping algorithm by intelligently choosing the threshold $\alpha$ to approximately preserve multivariate correlations or univariate means of large subsets of the data.

Takemura [242] presents an algorithm that combines local generalization and data swapping. The idea is to find, for every record, its "nearest record" based on a schema specific distance metric and swap/locally generalize the differing attributes. However, care should be taken that the pairing is two-way; i.e., if $x$ is paired to $y$, then $y$ should also be paired with $x$. Hence, Takemura proposes an algorithm based on maximum weight matching using the Edmonds Algorithm [89].

Primitive swaps were shown to be important also for the problem of generating entries in a contingency table given a fixed set of multiple marginals. Diaconis and Sturmfels [72] proposed a general algorithm for sampling from the set of tables that satisfy the fixed marginals. Their technique is very powerful since it can be used irrespective of what the marginals are. However, the technique's applicability is limited since it requires access to a *Markov basis*, or a finite set of "moves" such that any two tables that satisfy the marginal can be connected via a finite set of moves. Diaconis and Sturmfels use a Gröbner basis to construct their set of moves, but computing it even for tables with three dimensions is difficult. However, Dobra [75, 76] showed that when the marginals satisfy a property called *decomposability*, then Dalenius and Reiss' primitive swaps precisely form the Markov basis; i.e., every pair of tables satisfying a decomposable set of marginals is connected via a sequence of primitive swaps. Fienberg and McIntyre [101] present a great review of other work in data swapping and its variants. None of the above techniques are associated with formal privacy guarantees.

Controlled tabular adjustment (CTA) [48, 57, 63, 64] is another technique for perturbing a contingency table while preserving marginal statistics. The problem is formulated as an integer programming problem with linear constraints for the statistics to be preserved as well as for privacy (e.g., lower and upper protection levels on the cell values), and the objective function is a norm ($L_1$ [57, 63], $L_2$ [48], or $L_\infty$ [48]). Algorithms for CTA solve this integer problem either using an LP relaxation followed by rounding or using the interior point method. Privacy is analyzed in terms of whether the adversary can solve for the noise added to the cells of the table using a related integer program; but the authors consider privacy to be breached only when the adversary can determine the noise for a sufficient number of cells.

### 4.2.5 Model-Based Synthetic Data Generation

Another paradigm for releasing sanitized data sets is to generate a statistical model from a noise-infused version of the existing data, and to sample data points from this model to create the synthetic data set. Noise is introduced into the synthetic data from two sources: the noise infused prior to building the model and the noise due to random sampling.

The use of *multiple imputation* [223] to create synthetic data was first proposed by Rubin [221]. Under this proposal, a model is built over the data and then *multiple* data sets are sampled from the model (Liew et al. [159] had earlier proposed building a model of the data and sampling one data set from it). A variant of this approach, using multiple imputation to create *partially synthetic data*, has also been popular [1, 2, 100, 140, 159, 161, 182, 218]. To create partially synthetic data, a data publisher suppresses the sensitive attributes and then multiply imputes them from a model. Both of these multiple imputation approaches create multiple data sets which have the following benefits. First, a statistical model can be built on each of the data sets using standard software packages. These models can then be combined into a single model (for more details, see Section 5.3). This allows for better estimation of the variance of parameters of interest than if only one data set was released.

The choice of model is very important. An overly simplistic model will create a severe bias in the data. A model should be as general as possible and, ideally, nonparametric. For example, Reiter [218] studied the use of decision trees for creating partially synthetic data and Machanavajjhala et al. [167] used a multinomial model with a Dirichlet prior with one parameter for each element in the domain of the data. Polettini [203] presented an approach where different parts of the data were modeled differently. For each part, given a set of constraints on a set of variables, the maximum entropy distribution is constructed. Generalized linear models [175] are then used to model variables that do not appear in any constraints. Another common approach is to use algorithms like MICE [258] or Sequential Regression Multivariate Imputation (SRMI) [206] which generally work as follows. First, select an attribute whose values are to be imputed and learn a model for it based on the other attributes. Then, replace the value of the attribute with random samples from the predictive distribution according to the model. Then, choose another attribute whose values are to be imputed and learn a model for it based on the other attributes (including the attribute whose values were imputed in the previous step). Again, replace the true values with sampled values from the model. This process can be repeated several times for each attribute. See [1, 218] for more detailed examples of this process.

There are several reasons why multiply imputed data sets should be released instead of the actual models. In the case of MICE and SRMI, there are no explicit model parameters since the output data sets are essentially created using Gibbs sampling. Thus in this case there is no explicit model to release. In other cases, such as [167], the release of model parameters will result in a privacy breach. Thus if there is an explicit model and the parameters of the model cannot be used to breach privacy, then releasing the model is preferable to releasing data sampled from it.

An alternative approach to multiple imputation for creating partially synthetic data uses the concept of sufficient statistics. A *statistic* is a set of values computed from the data. A set of statistics is *sufficient* for a statistical model if that model can be built using only those statistics and without access to the original data. More formally,

a statistic is sufficient if the distribution of the data conditioned on the statistic is independent of the model parameters. For more details see [47]. To create sufficiency-based synthetic data, one first constructs a model (that admits sufficient statistics) for predicting the sensitive attributes given the non-sensitive attributes. Each sensitive attribute value is replaced by sample from the predictive distribution in such a way that the sufficient statistics are exactly preserved. For linear models, such constructions are given by [44, 183].

Franconi and Stander [104] present a different approach where a sensitive value is replaced by the midpoint of its predicted confidence interval if the original value was not one of the $p\%$ highest or $p\%$ lowest values in the data. Otherwise, if the original value is one of the $p\%$ lowest, it is replaced with a value higher than the predicted midpoint and if the original value is one of the $p\%$ highest, it is replaced with a value lower than the predicted midpoint.

Note that the sufficiency approach can also be used to replace the entire data set instead of just the sensitive values. Mateo-Sanz et al. [174] present a fast method for generating synthetic data that preserves the mean and covariance matrix of the data. Dandekar et al. [65] use ideas from Latin Hypercube Sampling to generate synthetic data with the aim of preserving univariate statistics as well as the rank correlation matrix.[4]

Different algorithms for generating synthetic data can be created by varying the synthetic model that is built using the data. However, these approaches in general do not have formal privacy guarantees because they do not analyze the disclosure caused by the use of estimated model parameters in the sampling process. For example, a single outlier can have a large influence on these parameters and the existence of an outlier in the original data may then be detected from the synthetic data and thus information about the magnitude of the outlier can be leaked. In some cases, the privacy guarantees require the assumption that the actual data really were generated exactly from a statistical model [182] even though it is widely recognized that "essentially, all

[4] Rank correlation [121] measures the correlation between the ranks of two variables instead of their actual values. The rank correlation is less sensitive to outliers than the standard covariance and correlation.

models are wrong, but some are useful" [38, p. 424], (see also Polettini and Stander's [204] response to Muralidhar and Sarathy [182]).

One simple technique that does provide formal privacy guarantees for synthetic data is the work of Machanavajjhala et al. [167] based on Dirichlet resampling. Let $U$ be a table with discrete attributes and let $D_U$ be its domain. Let $H$ denote the histogram of $U$, i.e., $H = \{f(v) \mid v \in D_U, f(v) =$ multiplicity of $v$ in $U\}$, and let $R$ denote a histogram of noise. Synthetic data are generated as follows. First, form the prior $Dir(H + R)$, where $Dir$ denotes the Dirichlet distribution. Then, draw a vector of probabilities, $X$, from $Dir(H + R)$, and generate $m$ points according to the probabilities in $X$. The above process is mathematically equivalent to the following resampling technique. Consider an urn with balls marked with values $v \in D_U$ such that the number of balls marked with $v$ equals the sum of the frequency of $v$ in $U$ and the frequency of $v$ in the noise histogram. Synthetic data are generated in $m$ sampling steps as follows. In each sampling step, a ball, say marked $v$, is drawn at random and two balls marked $v$ are added back to the urn. In this step, the synthetic data point is $v$.

Machanavajjhala et al. [167] characterized the privacy guaranteed by this algorithm in terms of the noise histogram. Specifically, they showed that in order to guarantee $\epsilon$-differential privacy, the frequency of every $v \in D_U$ in the noise histogram should be at least $m/(e^\epsilon - 1)$. For large $m$ and small $\epsilon$ the noise required for privacy overwhelms all of the signal in the data and renders the synthetic data completely useless. Such a large requirement of noise is due to the following worst case requirement of differential privacy. Consider a scenario where an adversary knows that $U$ contains exactly one record $u_i$ that can take either the value $v_1$ or $v_2$. Now suppose that in the output sample, every record takes the value $v_1$. If $m$ is large, then the adversary's belief that $r_u = v_1$ is close to 1 (see [167] for details). In order to guard against such scenarios, differential privacy requires large amounts of noise. However, the probability that such synthetic data are output is negligibly small, and so it may not always be necessary to guard against events which almost certainly will never happen. Thus this situation can be remedied using a weaker $(\epsilon, \delta)$-probabilistic differential privacy definition, where an algorithm is private if it satisfies $\epsilon$-differential privacy for all

outputs that are generated with a probability at least $(1 - \delta)$. Under this weaker definition, the Dirichlet resampling technique is private with much smaller noise requirements.

Rastogi et al. [211] propose another algorithm for synthetic data called the $\alpha\beta$ algorithm (also with privacy guarantees). It is similar to the select-a-size randomization operator [93] for publishing itemsets. Given an itemset $I$ that is a subset of the domain of all items $D$, the $\alpha\beta$ algorithm creates a randomized itemset $V$ by retaining items in $I$ with probability $\alpha + \beta$ and adding items in $D \setminus I$ with probability $\beta$. This algorithm satisfies a variant of $\epsilon$-differential privacy. Moreover, the authors show that for queries $Q : 2^D \rightarrow \mathcal{R}$, $Q(I)$ can be estimated as follows:

$$\hat{Q}(I) = (Q(V) - \beta Q(D))/\alpha, \tag{4.13}$$

where $Q(V)$ and $Q(D)$ are the answers to the query $Q$ on the randomized itemset $V$ and the full domain $D$, respectively. $\hat{Q}(I)$ is shown to provably approximate $Q(I)$ with high probability.

## 4.3 Summary

Privacy researchers have developed a plethora of sanitization algorithms; but, only a few of them satisfy provable guarantees of privacy. Much early research focused primarily on generalization and suppression-based techniques for privacy. However, these deterministic mechanisms do not guarantee privacy against powerful adversaries with probabilistic knowledge, thus increasing the significance of randomized solutions to the privacy problem.

# 5

# Using Sanitized Data

> Errors using inadequate data are much less than those using no data at all.
>
> — Charles Babbage

Once sanitized data are made public, if we are an attacker, then we proceed to Section 6. Otherwise, we must resolve the question of how to make good use of data, given that they have been sanitized. A recipient of sanitized data generally needs to answer two important questions. The first question is how much information has been lost due to the sanitization process, and the second question is how to perform the intended analysis using sanitized data. Utility metrics discussed in Section 3 aim to quantify the amount of information loss. In this section, we discuss how to use the sanitized data. In Section 5.1, we discuss how to answer queries over sanitized data. Since there is inherent uncertainty in the published data, the queries will necessarily have a probabilistic interpretation. In Section 5.2, we discuss data analysis over sanitized data from the point of view of machine learning and data mining, and in Section 5.3 we discuss statistical data analysis techniques for understanding whether a finding in the sanitized data

is statistically significant. We note that a detailed survey of privacy-related query processing, machine learning, data mining, and statistical techniques is beyond the scope of this paper. The purpose of this section is only to motivate the problems and point the readers to the related literature.

## 5.1 Query Processing

After applying sanitization techniques to the original data sets to enforce privacy criteria, the resulting sanitized data sets are usually imprecise (e.g., some attribute values have been generalized) or probabilistic (e.g., attribute values have been perturbed with random noise). For example, after we generalize Table 1.1 to enforce 4-anonymity and obtain Table 1.3, we are no longer able to distinguish between individuals' ages. On one hand, this imprecision is an unavoidable cost of privacy protection. On the other hand, a data user still wants to be able to answer queries about ages (e.g., the number of patients who have cancer and are less than 40 years old) as accurately as possible. In this section, we start with a brief introduction of general query processing for imprecise and probabilistic data and then cover techniques tailored to specific sanitization algorithms. Note that query processing for imprecise and uncertain data is an active and extensive research area in its own right; an exhaustive discussion is beyond the scope of this article.

**Probabilistic query processing:** One approach to querying a sanitized data set is to represent the data set in terms of a probabilistic database. Query processing techniques have been well-studied, and can be useful for answering queries on sanitized data sets interpreted in this manner. For example, Table 1.4 is a bucketized version of Table 1.1. A possible representation of such a table as a probabilistic database is shown in Table 5.1. This table can be interpreted to mean that each row appears with probability 0.5. Of course, this translation does not preserve all information from the bucketization; for example, it does not capture the fact that Ann has either heart disease or viral infection, but not both. More complex probabilistic data models such as those incorporating lineage can potentially be used to capture this

Table 5.1. Probabilistic representation of Table 1.4.

| | Age | Gender | Zip code | Nationality | Condition | Probability |
|---|---|---|---|---|---|---|
| Ann | 28 | F | 13053 | Russian | Heart Disease | 0.5 |
| Ann | 28 | F | 13053 | Russian | Viral Infection | 0.5 |
| Bruce | 29 | M | 13068 | Chinese | Heart Disease | 0.5 |
| Bruce | 29 | M | 13068 | Chinese | Viral Infection | 0.5 |
| Cary | 21 | F | 13068 | Japanese | Heart Disease | 0.5 |
| Cary | 21 | F | 13068 | Japanese | Viral Infection | 0.5 |
| Dick | 23 | M | 13053 | American | Heart Disease | 0.5 |
| Dick | 23 | M | 13053 | American | Viral Infection | 0.5 |
| | ... | ... | ... | ... | ... | |

[a]Records in Buckets 2 and 3 are omitted.

information if it is desired. After representing sanitized data sets in terms of a probabilistic database, we can apply efficient query processing techniques (e.g., [62]) to answer queries about information in the sanitized data sets, possibly after linking them with other public data (which can also be represented in the probabilistic database). Research projects that extend database query processing capability to uncertain or probabilistic data include [247, 249, 250, 251].

**OLAP on uncertain and imprecise data:** In a series of papers [42, 41, 43], Burdick et al. studied how to perform OLAP (online analytic processing) aggregate queries on imprecise or probabilistic data, where in the input data, each attribute of a data record to be aggregated can be a node in a generalization hierarchy or a probability distribution. Different records may be generalized to different granularities. The goal is to answer aggregate queries (e.g., SUM, COUNT, AVERAGE, etc.) over groups of any granularity. Burdick et al. discuss the desired properties of such aggregate queries and provide efficient algorithms. We note that this work can be used to answer aggregate queries and perform OLAP analysis over sanitized data sets generated by the generalization mechanism.

**Techniques specific to sanitization mechanisms:** A number of techniques have also been proposed to answer query based directly on the mechanisms used for sanitization. Agrawal et al. [17] use a technique similar to randomized response (see Section 4.2.1) to sanitize data and then present algorithms for answering aggregate range queries. Rastogi et al. [212] present a different sanitization mechanism in which a subset

of the data is retained and additional fake data records are inserted. They then show how to answer count queries over an arbitrary subset of the domain (not just axis-parallel range queries). The complexity of this algorithm depends on the complexity of the query (see [211] for full details of the query processing algorithm). In the case of the bucketization mechanism for sanitizing data (see Section 1.5), LeFevre et al. [150] and Zhang et al. [282] discuss how to provide upper and lower bounds on answers to aggregate queries.

## 5.2 Machine Learning and Data Mining

When the primary reason for data publication is to build machine learning or data mining models, the most direct measure of data utility is the accuracy (error rate or application-dependent metrics) of such models built on sanitized versions of data. In particular, one can compare the accuracy of a model built on the sanitized data with the accuracy of a model built on the original data to understand the amount of information loss (for the target application) incurred by privacy protection [107, 132, 151].

We also note that the sanitization process often destroys some structure of the original data, and sanitized data may have a different format than the original data. Thus, learning or mining algorithms may need to be adapted in order to be applied to the sanitized data. In this section, we first describe a recommended methodology for evaluating the utility of a sanitized data set for a specific purpose and then point the readers to a number of techniques for building models on sanitized data. A detailed discussion on how to build machine-learning models or how to apply data mining techniques to sanitized data is beyond the scope of this paper. In particular, we do not try to cover privacy-preserving data mining. See [12] for a detailed survey of this area.

### 5.2.1 Evaluation Methodology

To avoid obtaining overly optimistic accuracy estimates, strict training/testing data separation is recommended [151]. Before applying any sanitization technique to a data set $D$, one should first split the data set into two parts: the training set $D_1$ and the test set $D_2$, and then

apply a sanitization method to the training set *without any access to the test set*. Then one should build a model using only the sanitized training set $D_1^*$. Finally, one should evaluate the accuracy of the model using the test set $D_2$ by comparing the predicted target value (also called class label or response) with the actual target value of each case in the test set. Accuracy may be replaced by any application-dependent model-performance metric, and one can repeat this process multiple times and average model-performance numbers over different training/testing splits (e.g., an $n$-fold cross-validation). The important point is that, before testing a model, the test set should not be touched in any way. For example, any statistics computed from the entire original data $D$ should not be used in data sanitization, model construction or parameter tuning, because the test set $D_2$ is contained in $D$. This strict training/testing data separation is a standard practice in machine learning to prevent over-estimation of the performance of an algorithm.

### 5.2.2 Learning from Sanitized Data

We now review a number of useful techniques for building models on sanitized data.

**Learning from a generalized table:** Suppose that the sanitized data set $D_1^*$ is produced by applying a generalization algorithm to $D_1$ and the goal is to predict the target values of records in $D_2$ (without generalization). In this case, the training data $D_1^*$ are encoded using nodes in generalization hierarchies, but the attribute values in the test data $D_2$ are all at the leaf level of those hierarchies. One simple approach to handle this mismatch is to sample leaf-level values for records in $D_1^*$ that have non-leaf values. For example, let Table 1.2 be the generalized table $D_1^*$. For the first record, we can randomly sample an age value between 20 and 29, a gender value from {M, F}, a zip code between 13000 and 13099, etc. By repeating this procedure for each record, we can create a training set that has the same schema as $D_1$. This simple sampling method can serve as a baseline method for many other sanitization mechanisms. In [151], LeFevre et al. studied a range-encoding method. This idea is best-suited to numeric data, and simply replaces each generalized value with two separate features representing

the upper and lower bounds of the range. Other methods include decision tree learning [279], Naive Bayes learning [280], and rule learning [130] for hierarchical attribute values.

**Learning from noisy data:** Suppose that the sanitized data set is generated by adding noise to the attribute values of the records in the original data set. Learning from such sanitized data is generally called learning from noisy or uncertain data in the literature. Usually the noise distribution is assumed to be known. For a general introduction to this problem in linear regression, see Section 9.6 of [231]. Recent work includes methods for decision tree induction [16], Bayesian regression [252], support vector machine classifiers [34, 233], and nearest neighbor classifiers [8].

**Learning from group statistics:** Suppose that the sanitized data set is generated by first grouping or bucketizing records of the original data set and then releasing summary statistics for each group. Here, the data user can build a model using only these statistics. Learning methods for this setting include [67, 185, 205]. This setting is also related to the multiple instance learning problem, e.g., [22, 73].

**Learning from multiple views:** Suppose that the sanitized data set $D_1^*$ consists of multiple aggregate views of the original data set $D_1$, each of which contains a subset of the attributes of records in $D_1$ and summary statistics (e.g., COUNT, AVERAGE, etc.) for each distinct combination of the values of these attributes. For count views, which are commonly known as marginals (or marginal contingency tables), iterative proportional fitting [35] is the classic method for estimating information about the original data table $D_1$. In recent work, Chen et al. [50] developed an ensemble-based method for building classifiers on $D_1^*$. For a Bayesian perspective, see [75].

## 5.3 Statistical Analysis

While the database, data mining and machine learning literature provide methods to answer queries over sanitized data and build models from the data, these techniques usually do not address whether a finding from the data is statistically significant or not. For example, one may use a query processing technique to compute the sample mean;

however, to determine whether this value is significantly different from a null hypothesis, one also needs to estimate the variance and/or confidence intervals. We will discuss a general model for statistical analysis of sanitized data due to Little [160] in Section 5.3.1 and then we will discuss specialized analysis for multiply imputed (partially) synthetic data in Section 5.3.2.

### 5.3.1 A General Model

In this section we will use the words "masked" and "sanitized" interchangeably. Little [160] provided a general framework for performing statistical analysis over masked (sanitized) data which builds upon statistical models for missing and coarsened data. In this model there is an $n \times p$ *data matrix* $\mathbf{X}$ where each row represents a data item (such as an individual) and each column represents an attribute, so that $x_{ij}$ represents the measured value of attribute $j$ for individual $i$. The output data are represented with the help of two $n \times p$ matrices $\mathbf{M}$ and $\mathbf{Z}$ called the *masking indicator matrix* and *masking treatment matrix*, respectively. We will use the notation $m_{ij}$ and $z_{ij}$ to represent the value in row $i$ and column $j$ for the matrices $\mathbf{M}$ and $\mathbf{Z}$, respectively. The value of $m_{ij}$ is 1 if the value of $x_{ij}$ has been sanitized and 0 otherwise. The value of $z_{ij}$ is the masked value of $x_{ij}$ if $m_{ij} = 1$ and is unobserved otherwise. Note that $z_{ij}$ could be a number, a code indicating a missing value, etc. It is helpful to partition $\mathbf{X}$ into two parts $(\mathbf{X}_{obs}, \mathbf{X}_{mis})$ where $\mathbf{X}_{obs}$ is the subset of $\mathbf{X}$ that is observed in the output and $\mathbf{X}_{mis}$ is the subset of $\mathbf{X}$ for which the values are missing. It is also convenient to partition $\mathbf{Z}$ into $(\mathbf{Z}_{obs}, \mathbf{Z}_{mis})$ where $\mathbf{Z}_{obs}$ corresponds to values observed in the output and $\mathbf{Z}_{mis}$ corresponds to missing values $\mathbf{Z}$. Note that $\mathbf{M}$ and $\mathbf{Z}$ may not be known to the analyst and in this case they would need to be treated as random variables.

The primary statistical interest is in estimation and hypothesis testing for a parameter $\theta$ which corresponds to a hypothetical model that generated the data. Thus the data distribution is $f_X(\mathbf{X}|\theta)$. The masking treatment matrix $\mathbf{Z}$ depends on the data matrix through the distribution $f_Z(\mathbf{Z}|\mathbf{X})$ and the masking indicator matrix depends on the rest through the distribution $f_M(\mathbf{M}|\mathbf{Z}_{obs}, \mathbf{X})$ (according to Little [160],

the masking indicator matrix should not depend on the missing values $\mathbf{Z}_{mis}$). Inference can be based on the likelihood function for $\theta$ which is obtained by integrating out the unobserved data:

$$\mathcal{L}(\theta \,|\, \mathbf{M}, \mathbf{X}_{obs}, \mathbf{Z}_{obs}) = \int f_M(\mathbf{M} \,|\, \mathbf{Z}_{obs}, \mathbf{X}) \; f_Z(\mathbf{Z}_{obs} \,|\, \mathbf{X}) \times f_X(\mathbf{X} \,|\, \theta) \; d\mathbf{X}_{mis} \tag{5.1}$$

In case the masking indicator matrix $\mathbf{M}$ is unknown, one would also have to integrate out the uncertainty in $\mathbf{Z}_{obs}$ and $\mathbf{M}$.

Little [160] points out additional ways this likelihood can be simplified. If $f_M(\mathbf{M} \,|\, \mathbf{X}, \mathbf{Z}) \equiv f_M(\mathbf{M} \,|\, \mathbf{X}_{obs}, \mathbf{Z}_{obs})$ then the masking selection mechanism is said to be ignorable. If $f_Z(\mathbf{Z}_{obs} \,|\, \mathbf{X}) \equiv f_Z(\mathbf{Z}_{obs} \,|\, \mathbf{X}_{obs})$ then the masking treatment mechanism is said to be ignorable. Using these notions, Little derives simplified versions of the likelihood function. If the masking selection mechanism is ignorable then we can use:

$$\mathcal{L}(\theta \,|\, \mathbf{M}, \mathbf{X}_{obs}, \mathbf{Z}_{obs}) = \int f_Z(\mathbf{Z}_{obs} \,|\, \mathbf{X}) f_X(\mathbf{X} \,|\, \theta) \; d\mathbf{X}_{mis}$$

If the masking treatment mechanism is ignorable then we can use:

$$\mathcal{L}(\theta \,|\, \mathbf{M}, \mathbf{X}_{obs}, \mathbf{Z}_{obs}) = \int f_M(\mathbf{M} \,|\, \mathbf{Z}_{obs}, \mathbf{X}) f_X(\mathbf{X} \,|\, \theta) \; d\mathbf{X}_{mis}$$

If both are ignorable then we can use:

$$\mathcal{L}(\theta \,|\, \mathbf{M}, \mathbf{X}_{obs}, \mathbf{Z}_{obs}) = \int f_X(\mathbf{X} \,|\, \theta) \; d\mathbf{X}_{mis}$$

(in which case the analysis only depends on the values that have not been sanitized). More details on how to apply this analysis can be found in [160]. A detailed analysis for data perturbed by multiplicative noise is presented by Hwang [129] and another measurement errors model is also discussed by Fuller [106].

### 5.3.2 Statistical Analysis of Multiply Imputed Synthetic Data

In this section we focus on a promising approach for the analysis of synthetic data generated by multiple imputation (see Section 4.2.5). Multiple synthetic data sets are generated from a model and ideas based

on the theory of multiple imputation [223] are used to measure the variance of an estimator computed from the sanitized synthetic data. The benefit of this approach is that off-the-shelf statistical analysis tools can be used and then a correction defined in terms of a combination rule outputs the desired result.

In order to describe how to analyze this kind of sanitized data, we first provide a brief discussion of multiple imputation outside the context of privacy. Multiple imputation (MI) is a tool developed by Rubin [223] for handling data sets that contain missing values. The basic idea of MI is to create multiple complete data sets by filling in missing values by sampling them from a model built from the rest of the data. An alternative approach is to fill in each missing value with a single value (for example, by using a maximum likelihood method). However, this has the drawback of causing statistical software to underestimate the variance in the data because each filled-in value is treated in the same way as a value that is actually present in the data [223]. Multiple imputation comes with a simple estimation procedure that helps avoid this problem.

The estimation of parameters from a multiply imputed data set proceeds as follows. Suppose we have used multiple imputation to create $m$ data sets $D_1, \ldots, D_m$. On each data set we compute a population statistic $Q_i$ (say, the sample mean) and $U_i$ (the estimate of the variance of $Q_i$). We then compute the overall average $\bar{Q} = \frac{1}{m}\sum_{i=1}^{m} Q_i$, the average estimated variance $\bar{U} = \frac{1}{m}\sum_{i=1}^{m} U_i$, and the between-sample variance $B = \frac{1}{m-1}\sum_{i=1}^{M}(Q_i - \bar{Q})^2$. We then return $\bar{Q}$ as the estimated population statistic (say, the mean of the population) and

$$\bar{U} + \frac{m+1}{m}B \tag{5.2}$$

as the estimate of the variance of $\bar{Q}$. This is known as the *combination rule*. For confidence intervals, significance levels, and other statistical computations over multiply-imputed data, see the book by Rubin [223] and also [222, 228, 224, 20, 156, 209, 176].

In the context of privacy, the analysis of partially and fully synthetic data (that are repeatedly sampled from a model) relies on combination rules similar to the ones for multiple imputation (such as

Equation (5.2)). These combination rules typically contain additional additive terms that account for variability due to the sanitization process, but these terms are based on large data assumptions and may not give correct answers for small data sizes. Raghunathan et al. [207] present a combination rule for purely synthetic data, and Reiter [217] shows how to perform hypothesis testing. Reiter [213] also presents combination rules for partially synthetic data. Finally, Reiter [215] shows how to perform inference from data where multiple imputation was used both to fill in missing data and to sanitize the data.

## 5.4 Summary

We believe that data analysis remains one of the key challenges in the area of privacy-preserving data publishing. Related work in the areas of probabilistic query processing, Machine Learning, and Statistics provides techniques that are useful for analyzing sanitized data. However, in comparison to research in the production of sanitized data, work in this field is relatively sparse. Thus, research that addresses the issues specific to sanitized data using rigorous methodology appears to be a fruitful area of future work.

# 6

# Attacking Sanitized Data

> Hence that general is skillful in attack whose opponent does not know what to defend.
>
> — Sun Tzu (Art of War)

Data are sanitized prior to publication so that sensitive information is not presented in the clear (according to some privacy criterion). In some cases, however, an attacker can still glean sensitive information from the sanitized data using varying amounts of skill, creativity, and effort.

In practice, an organization assessing the sensitivity of data to attacks needs to consider a variety of social issues. Elliot and Dale [90] categorize these issues in terms of:

- Motivations/Goals — these include determining if an individual is in a data set, publicizing sensitive values belonging to an individual, embarrassing an organization, and using sensitive information for fraud.
- Means available to an attacker — these include the attacker's statistical skills, external knowledge, and availability of computational resources.

- Opportunity — the ease of access to the data (is it publicly available or are the data housed in a secure computing facility?).
- Attack Type — this includes verifying information about specific individuals (including verifying their presence in the data), discovering information about individuals that are outliers in the sanitized data, making a claim that a breach is possible, or partially reconstructing the original data.
- Presence of Matching Variables — attributes that can be used for linking with external data/knowledge need to be identified.
- Target Variables — sensitive attributes like disease or salary need to be identified. The degree of sensitivity of these attributes also needs to be assessed.
- Effects of noise or errors in the data — noise may already reduce the chance of a successful attack [36].
- Likelihood of attack and of achieving attack goal.
- Alternative means to achieving this goal — for example, by using a subpoena or by physical surveillance.
- Consequences to the organization of the attempted attack — this includes loss of trust and potential lawsuits.

Here we are mainly interested in issues concerning *attack type*, i.e., identifying that an individual is in the data, reconstructing information about individuals, etc.

An attack against a sanitized data set can be considered a success if an attacker, through reasonable means, *believes* that he or she has discovered sensitive information. This view is fairly common in the literature (see, for example [98, 106, 148]) and necessitates the modeling of potential attackers. Although the attacker is required to use reasonable means, the actual reasonableness of a procedure is subject to interpretation. For example, Kifer [141] has shown that random worlds reasoning [25, 118], which is commonly used for modeling attackers, has some unreasonable properties and that alternative models can yield more accurate reconstruction of sensitive information. Nevertheless, random worlds is an intuitive model that is likely to be used by less sophisticated attackers.

The actual sensitive information that the attacker has deduced may be a property of an individual (or tuple in a table) or it may be an aggregate property of a subset of the data. An example of the former would be the statement "Cary has AIDS," and an example of the latter would be the statement "the average daily search volume of internet giant Yagosoft is approximately 300 queries". Note that these statements can be problematic *even if they are not correct*. If an attacker can convincingly reason that "Cary has AIDS" with 50% confidence (even though Cary does not have AIDS), this may cause an insurer to reject Cary's application for health insurance. Similarly, an advertiser may reconsider its relationship with Yagosoft, instead signing a contract with its rival Microhoogle, based on a perception of user popularity.

We gave a short history of real-world attacks data in Section 1.2. In this section, we will discuss attack strategies that have been proposed in the literature. These strategies fall into roughly two categories: direct attacks on the sanitization schemes (Section 6.1) and attacks that leverage external information (Section 6.2). We would like to clarify that while some of the attacks we describe are designed for sanitization algorithms that do not satisfy formal notions of privacy (for example, see Section 6.1.1), other attacks are designed for sanitization algorithms that do satisfy formal privacy definitions (in fact, they take advantage of assumptions made by those definitions).

In section 6.1, we will discuss attacks due to (a) heuristic notions of privacy (Section 6.1.1), (b) knowledge of algorithmic details (Section 6.1.2), (c) alternate forms of reasoning (Section 6.1.3), (d) latent structure in the data (Section 6.1.4), and (e) undesired uses of the data (Section 6.1.5).

Since an attacker's side information is also a concern, a data publisher needs to understand what kinds of knowledge attackers have access to, and how this knowledge can be utilized to attack data. In Section 6.2, we will discuss (a) how attackers use external data for linking attacks (Section 6.2.1), (b) attacks due to composing multiple sanitized data sets (Section 6.2.2), (c) attacks that derive knowledge from similar data (Section 6.2.3), and (d) attacks due to instance level background knowledge (Section 6.2.4).

## 6.1 Attacks on Sanitization Schemes

In this section we discuss direct attacks on sanitization schemes.

### 6.1.1 Combinatorial Methods

One of the earliest attacks on a sanitization system was described by Schlorer [230, 71] and is called a *tracker*. It was used to show the vulnerability of certain query answering systems. Although it was not an attack against published data, per se, it developed ideas that were later used to demonstrate attacks on published data. The query answering system had a database of $n$ records and a safety parameter $k$. It was only allowed to answer COUNT and SUM queries (e.g., "What is the total salary of female employees?" and "How many employees were younger than 30 in 1997?") as long as the number of tuples used in the computation of the answer was between $k$ and $n - k$. Schlorer showed that if an attacker knows some unique characteristics of a target individual, then by posing a few carefully chosen queries and using linear algebra, additional characteristics (such as salary) of the target individual can also be determined. For example, if the attacker knows that Shirley is the only female under 30 in the data set, the attacker can pose the queries "what is the total salary of employees under 30" and "What is the total salary of male employees under 30?" to determine Shirley's salary. In this case, the tracker was the query "what is the total salary of male employees under 30" since it did not contain information about Shirley but "helped track down additional characteristics" [70] of her record.

The combinatorial flavor of the tracker attack can be seen in the following data publishing problem. Suppose a data publisher who is in charge of medical records with attributes {*Age*, *Gender*, *Zip code*, *Nationality*, and *Condition*} is considering whether or not to publish several histograms (also called *marginals* in the official statistics literature) of the data. For example, the data publisher might consider publishing one histogram on *Gender* and *Nationality* and another histogram on *Gender* and *Condition* (see Table 1.6 for example histograms). Each bucket of each marginal histogram (e.g., (Male, US) and (Male, AIDS)) may have a count of at least $k$, but at the same

time there are dependencies between the different histograms due to which some buckets in the complete table (e.g., (45, Male, 1480, US, AIDS)) may have counts less than $k$.

Dobra [75] considered this scenario, and he showed, for each combination of attributes in the original table, how to compute combinatorial upper and lower bounds on the number of tuples in the table with those particular attributes. While this technique allows inference about the number of people with a particular set of attributes, sometimes the data publisher is interested in possible inferences about the sensitive values of an individual. General probabilistic inference using Markov Chain Monte Carlo [220] techniques was discussed by Diaconis and Sturmfels [72] and Dobra [75]. These techniques use the notion of a Groebner Basis, a fundamental tool in algebraic geometry [56]. Also, Yao et al. [278] present a technique for analyzing multiple views of the same table to determine how many values of a sensitive attribute can be ruled out for an individual.

Another variation of this problem is to consider what inference is possible given histograms and conditional distributions (e.g., $P(Age \mid Zip\ code)$). Conditional distributions are useful for generating association rules and sometimes may reveal less information about individuals than a histogram. Based on such information, Slavkovic and Fienberg [235] and Fienberg and Slavkovic [102] show, for each combination of attributes in the original table, how to compute upper and lower bounds on the number of tuples in the table with those particular attributes.

### 6.1.2 Optimality Attacks

A different class of attacks uses the following observation. Usually data anonymization is framed as a constrained optimization problem: produce the table with the smallest distortion that also satisfies a given set of privacy requirements. Wong et al. [271] (and independently Fang and Chang [94]) presented the *minimality attack* to illustrate a danger of this approach. To perform this attack, an attacker will usually need to know the non-sensitive information of many individuals in the table, the privacy policy, and the algorithm used for anonymization. Consider Table 6.1(a), and suppose that the zip codes and genders of the patients

Table 6.1. A table subject to the minimality attack.

(a) Original Table

| Zip code | Gender | Disease |
|---|---|---|
| 94085 | M | HIV |
| 14085 | M | HIV |
| 14085 | F | None |
| 94085 | F | HIV |
| 14085 | F | Flu |
| 14085 | F | None |
| 14085 | F | None |
| 14085 | F | Flu |

(b) Sanitized Table

| Zip code | Gender | Disease |
|---|---|---|
| * | * | HIV |
| * | * | HIV |
| * | * | None |
| * | * | HIV |
| * | * | Flu |
| * | * | None |
| * | * | None |
| * | * | Flu |

are not sensitive but that the diseases are. Further, suppose the following privacy policy is desired: for each published combination of gender and zip code, at most half the corresponding patients have HIV. Clearly the original table cannot be published because of the male and female patients in zip code 94085. Suppose the anonymization algorithm acts by choosing a set of non-sensitive attributes and suppressing the values of these attributes for all tuples. The only safe sanitized table is the one shown in Table 6.1(b). An attacker who sees Table 6.1(b), and who knows the non-sensitive attributes of every individual in the table, will reason as follows: if all the HIV patients were female, or if exactly one of the HIV patients were male, then the privacy requirement could have been achieved by suppressing only the zip code. Therefore both male patients must have HIV. Similarly, if at most one of the patients from zip code 94085 had HIV, then the privacy requirement could have been satisfied by suppressing only gender. Thus both patients from zip code 94085 must have HIV.

Note that if an attacker does not know the non-sensitive attributes of every individual, there may be an external data set (such as a voter list) that lists this information. However, such an external data set may also contain information about many individuals not in the data set. For example, if the attacker had no external information other than a data set listing five males and five females from each zip code, then it would be harder to draw any conclusions from Table 6.1(b). This suggests that sampling may be a useful step in the anonymization process. Wong et al. [271] propose a different approach to guard against the minimality attack. The essential idea of this approach is to alter the

sensitive values of certain tuples. While this can reduce the correctness of the published data, it makes a minimality attack much more difficult.

Zhang et al. [281] present a general formal model for the minimality attack which can work with many privacy definitions and deterministic sanitization algorithms. Let $x$ be the data set that needs to be sanitized. In this model, a *disclosure schema* $T = (s_1, \ldots, s_{n_T})$ is a partition of the domain of the original data set $x$. There are many candidate disclosure schemas $T^1, \ldots, T^k$, and the goal of a sanitization algorithm $\mathcal{A}$ (on input $x$) is to first choose a disclosure schema (partitioning of the domain) $T^i$, and then to output the partition $s^i_j \in T^i$ that contains the original data set $x$.

As an example, suppose that the domain of the data set is $\{1,2,3,4,5,6,7,8,9\}$. Table 6.2(a) shows several possible disclosure schemas. For example, $T^3$ partitions the domain into three groups — the first group consists of the three data sets $1,2,3$, the second group consists of $4,5,6$, etc. A typical sanitization algorithm is shown in Table 6.2(b). This particular algorithm behaves as follows. If the original data set is 1, then it chooses the disclosure schema $T^1$ and returns the group from $T^1$ that contains the input data set (i.e., it returns $(1,2)$). If the original data set is 2, it chooses $T^2$ and returns $(1,2,3)$, etc. For what follows, assume that the original data set is 2, in which case the sanitization algorithm in Table 6.2(b) returns the output $(1,2,3)$.

Now, an attacker may know *a priori* that the original data set $x$ can belong to only a subset $D$ of the original domain. For example, in

Table 6.2. Sanitization algorithm and disclosure schemas for a data set whose domain is $\{1,2,3,4,5,6,7,8,9\}$.

(a) Disclosure Schemas

| ID | Disclosure Schema |
|---|---|
| $T^1$ | $\{(1,2),(3,4),(5,6),(7,8,9)\}$ |
| $T^2$ | $\{(1,2,3),(4,5),(6,7),(8,9)\}$ |
| $T^3$ | $\{(1,2,3),(4,5,6),(7,8,9)\}$ |

Sanitization Algorithm

| Input | Disclosure Schema | Output |
|---|---|---|
| 1 | $T^1$ | $(1,2)$ |
| 2 | $T^2$ | $(1,2,3)$ |
| 3 | $T^1$ | $(3,4)$ |
| 4 | $T^3$ | $(4,5,6)$ |
| 5 | $T^3$ | $(4,5,6)$ |
| 6 | $T^3$ | $(4,5,6)$ |
| 7 | $T^3$ | $(7,8,9)$ |
| 8 | $T^3$ | $(7,8,9)$ |
| 9 | $T^3$ | $(7,8,9)$ |

the real world, the attacker may know that males do not suffer from ovarian cancer and, in our running example, the attacker may know that the original data set is not 3. Thus the attacker knows that $x \in D$ and $x \in s_j^i$. In our running example, since the sanitization algorithm in Table 6.2(b) returned $(1,2,3)$, the attacker would know (from the output) that the original data set is either 1, 2, or 3 and will use background knowledge to rule out 3.

Additionally, given knowledge of a sanitization algorithm $\mathcal{A}$, an attacker would also know that the algorithm chose a particular disclosure schema $T^i$ in the first place. Let $L^i$ denote the collection of data sets $y$ such that on input $y$, algorithm $\mathcal{A}$ chooses the disclosure schema $T^i$. Thus the original data set $x$ must be an element of $L^i \cap s_j^i \cap D$. This subset of the original domain, which we denote by $DS_x$ is known as the *disclosure set* for $x$. In our example, the algorithm had chosen $T^2$ and output $(1,2,3)$. We are therefore interested in the value $L^2 = \{2\}$ which means that only input 2 could have caused the sanitization algorithm to select $T^2$. Therefore $DS_2 = \{2\}$, in which case knowledge of the sanitization algorithm revealed the original data set.

To measure privacy, there is assumed to be safety predicate $p$ whose input is a subset of the original domain and whose output is true or false. The algorithm $\mathcal{A}$ is *p-safe* if for all data sets $y$, $p(DS_y) = \text{true}$ (in our example, any reasonable $p$ would mark $DS_2$ as unsafe because it contained only one data set, and so the algorithm in Table 6.2(b) is not $p$-safe). A $p$-safe algorithm will thus protect against an attacker who believes that all data sets in a disclosure set are equally likely. Zhang et al. [281] also provide $p$-safe algorithms that try to optimize utility.

### 6.1.3 Alternative Reasoning

Kifer [141] presented an attack based on the observation that the sanitized data themselves can leak more information than the data publisher anticipated, even when an attacker does not have detailed knowledge of the sanitization algorithm. This problem can arise when an attacker reasons about the sanitized data in a way that is different (and perhaps more realistic) from the reasoning used by the data publisher. Thus it is important for a data publisher to reason about sanitized data

in several ways to simulate attackers with varying degrees of sophistication. To illustrate the problem of only considering one approach, we will briefly compare the *random worlds* model [25, 118], the *independence* model, and reasoning based on exchangeability [229].

The *random worlds* model (as applied to data privacy) models an attacker with no probabilistic preferences. Given a sanitized data set $\mathcal{S}$, there is a collection of input data sets $\mathcal{I}$ that could have caused the sanitization algorithm to output $\mathcal{S}$. Using the principle of indifference, the attacker considers each data set in $\mathcal{S}$ to be equally likely. Using this probability distribution, the attacker then reasons about a target individual using Bayes Theorem. This reasoning is used either explicitly or implicitly by much of the literature, including [51, 142, 168, 173, 271, 273, 281]. Du et al. [82] use a form of maximum entropy reasoning which is related to random worlds reasoning. In certain cases (such as unary predicates [118] or decomposable marginals [75]) the two models are exactly, asymptotically, or approximately identical, and the weakness of random worlds (discussed below) also applies to maximum entropy.

In the *independence* model, the attacker believes that each tuple in the database is generated independently by a some probability distribution (each tuple can have its own distribution). This model is also common [61, 91, 179, 212]. In some cases this is justified by the claim that the attacker may know the true data-generating probability distribution. Notice that this claim suffers from a weakness in that it makes assumptions about the data in order to guarantee privacy (in particular, it assumes that the data are generated independently). Thus this claim requires modeling both the data and the attacker. Note that the independence assumption is clearly violated in the case of a family member contracting a contagious disease and passing it along to the rest of the family; tuples corresponding to such families are not independent. Thus when modeling an attacker it is preferable to avoid modeling the data.

*Exchangeability* is a concept that generalizes independence and is one of the foundations of Bayesian reasoning. A sequence of random variables is exchangeable if any permutation of finitely many elements in the sequence results in a sequence that is equally likely. A deep result due to Bruno de Finetti states that if one believes a sequence

of random variables to be exchangeable then this is mathematically equivalent to believing that points in the sequence were generated independently given some unknown probability distribution that itself was chosen randomly. Thus an attacker who believes the original data were generated in this way does not have to treat the tuples independently; the attacker may believe that the tuples are correlated and that this correlation results from a shared, unknown distribution. For more details, see [141, 229, 255].

To see these models in action, consider Table 6.3 (reproduced from [141]). How would an attacker reason about the missing value for tuple 200? Using random worlds, it turns out that the attacker would believe that tuple 200 has cancer with probability 0.5 despite the fact that the table appears to show a strong correlation between lung cancer and smoking. In particular, the attacker's belief about tuple 200 is unaffected by the rest of the table. Similarly, an attacker using the independence model will not change his or her belief about tuple 200 after seeing the tuples 1–199. In fact, if the attacker believed that smoking cured cancer, then the attacker would attribute the apparent correlation (between smoking and cancer) displayed by Table 6.3 as a byproduct of pure chance. On the other hand, an attacker who believed in exchangeability would be able to learn about this correlation. Kifer [141] shows how an attacker who believes in exchangeability but has

Table 6.3. A data set related to smoking.

| Tuple ID | Smoker? | Lung Cancer? |
|---|---|---|
| 1 | n | n |
| 2 | n | n |
| ⋮ | ⋮ | ⋮ |
| 98 | n | n |
| 99 | n | n |
| 100 | n | n |
| 101 | y | y |
| 102 | y | y |
| ⋮ | ⋮ | ⋮ |
| 198 | y | y |
| 199 | y | y |
| 200 | y | ? |

Table 6.4. Dataset protected with bucketization.

| Tuple ID | Smoker? | GID |
|---|---|---|
| 1 | y | 1 |
| 2 | y | 1 |
| 3 | n | 2 |
| 4 | n | 2 |
| 5 | y | 3 |
| 6 | n | 3 |
| 7 | y | 4 |
| 8 | y | 4 |
| 9 | n | 5 |
| 10 | n | 5 |
| 11 | y | 6 |
| 12 | n | 6 |

| GID | Disease |
|---|---|
| 1 | Cancer |
| 1 | Flu |
| 2 | Flu |
| 2 | None |
| 3 | Cancer |
| 3 | None |
| 4 | Cancer |
| 4 | None |
| 5 | Flu |
| 5 | None |
| 6 | Cancer |
| 6 | None |

no other preference about smoking and cancer will end up with an increased belief that tuple 200 has cancer.

As a demonstration, Kifer [141] shows how reasoning based on exchangeability can be used to attack bucketization schemes such as Anatomy [273] (Section 4.1.4). Consider Table 6.4 (reproduced from [141]), which is a possible output of Anatomy. In this table, the tuples are partitioned into groups, a group id (GID) attribute was added, and a second table that lists the diseases for each group was constructed. Notice that the groups leak information about each other. For example, no group consisting entirely of nonsmokers has cancer as one of the diseases. Thus we may reason that tuple 12, which is a nonsmoker, belongs to a group that has exactly one cancer patient. Since the other tuple belongs to a smoker, we could conclude that tuple 12 is less likely to have cancer (while random worlds reasoning would give a probability of 0.5). This attack is formalized in [141], which also sketches attacks on several other sanitization schemes. Experimentally, it was shown that reasoning using exchangeability provided better inference of sensitive attributes than random worlds even when exchangeability was used to model an attacker with no prior preferences [141]. In addition to higher overall accuracy, this attack showed that some tuples are especially at risk since their estimated probability of having a particular sensitive value was very large (in those cases the predicted sensitive value is generally the same as the true sensitive value).

Finally, note that Evfimievski et al. [91] use very general classes of probability distribution over data sets known as log-submodular and log-supermodular distributions. These ideas are also investigated by Rastogi et al. [210]. Although there are no published demonstrations of how such a distribution can be chosen and successfully used to attack sanitized data, this is an interesting direction to pursue.

### 6.1.4 Denoising

While the preceding attacks were designed for anonymization schemes that produce "truthful" data (such as histograms), other attacks have been used against schemes that add noise. In this scenario, the attributes of the tuples are numeric. The simplest anonymization scheme simply adds independent Gaussian random variables (with the same variance) to each attribute of each tuple. Paass [195] found that this generally does not offer much protection against re-identification of individuals in the data as perturbed tuples can be linked to the original tuples. Kargupta et al. [138] and Huang et al. [126] also show how to remove noise from such data. Kim [143], Tendick [243], and Fuller [106] proposed using noise with covariance structure similar to that of the data (although this too can be attacked [80]), and many other new data perturbation schemes and attacks have also been proposed. For a survey of such attacks, see [162].

An important attack not covered in [162] is the use of linear programming [74, 88]. In this setting, suppose the database is an ordered list of $n$ items $\{d_1, \ldots, d_n\}$ where each item $d_i$ is either 0 or 1. A query is represented as a set $q$ of indices, and the answer to the query is the sum of the corresponding elements (i.e., $\sum_{i \in q} d_i$). Dinur and Nissim [74] consider the following question: suppose $n(\log n)^2$ queries are generated uniformly at random and the answer to each query is perturbed with arbitrary noise of magnitude $o(\sqrt{n})$. How well can the attacker reconstruct the original table? They showed that by using linear programming and rounding the result, an attacker can reconstruct the database up to $\epsilon n$ mistakes with probability $1 - \delta$ (for any $\epsilon > 0$, $c > 0$ and $\delta \in o(1/n^c)$, and a large enough $n$). Dwork et al. [88] later extended this work and showed that if weighted queries are allowed (i.e.,

$\sum_{i \in q} a_i d_i$), where the weights are generated according to the standard Gaussian distribution, then the answer to $O(n)$ queries cannot be published even if $pn$ (for $p < 0.239$) queries are arbitrarily wrong and noise bounded by $\alpha$ is added to the rest of the answers. Otherwise, with high probability (i.e., $1 - \delta$, where $\delta \in o(1/n^c)$ for all $c > 0$), the original database can be reconstructed with at most $O(\alpha)$ mistakes. Intuitively, these results say that the amount of statistically meaningful information in a data set is sub-linear in the size of the data. Note that in both cases the results are asymptotic, so they require a large enough $n$ in order for $\delta$ to be small.

For time series data, Papadimitriou et al. [198] propose using linear filters and linear regression (on points known to an attacker) to remove some of the variance caused by the addition of noise.

### 6.1.5 Undesired Uses of Data

An interesting attack described by Palley and Simonoff [196] is based on the notion that building a particular model over the data may be considered a violation of privacy. For example, if data about a company is released, an employee may build a statistical model on the data to predict wages. This employee may then compare his wages with his predicted wages, and would be upset if the true wages are less than they should have been according to the model. The company may therefore want to prevent anyone from building a regression model for wages from the data (e.g., building a statistical model for an attribute would be an undesired use of the data).

While this attack was described for a statistical (query answering) database, it can be extended to data publishing by considering the query answers to be the published data. Palley and Simonoff [196] demonstrated how to build a linear regression model from a database that only allowed count, average, and sum-of-squares queries over subsets of the data. The main idea is to first build a 1D histogram on each attribute, to then use these histograms to identify regions of the domain that should be queried, to construct artificial data sets that would give the same answers to such queries, and finally to create a linear regression model from such data sets. Palley and Simonoff [196]

found that, in general, this technique provides fairly accurate regression models compared to models built on the original data.

## 6.2 Attacks Using External Information

External information provides an attacker with additional tools for attacking a sanitized data set. In fact, all the attacks in Section 1.2 and the minimality attack in Section 6.1 heavily relied on external information. This information can come in the form of data sets, specific instance-level information about individuals in the data, or statistical knowledge.

### 6.2.1 Linking Attacks

The most common form of attack, the *re-identification* attack, uses record linkage techniques to link tuples in external data to sanitized data. The literature on record linkage is vast and has been studied in the statistics, artificial intelligence, and database communities. Winkler has provided several overview papers on the use of record linkage in the context of privacy [265, 268, 269, 270], and so we will briefly mention some related results.

Given two files $A$ and $B$ containing lists of tuples, the main idea of record linkage is to classify pairs $(a,b)$ (where $a \in A$ and $b \in B$) as match or non-match with various levels of uncertainty [96, 244]. Linkage is complicated by the fact that the files $A$ and $B$ may contain noise and data entry errors. Since the data sanitization process can be thought of as an infusion of noise and errors, the theory of record linkage fits very well in this scenario. Here $A$ can be treated as an external data set such as a voter registry and $B$ is the sanitized data set. The goal of a linkage or re-identification attack would be to associate a tuple from $A$ to a tuple from $B$ and later to use this information to derive better estimates of sensitive information about the individual(s) corresponding to these tuples. There are many techniques for performing this linkage, including Bayesian methods [99, 216], discriminant analysis [195], bipartite matching [153], and nearest neighbor methods [266, 269]. In some cases, linking to synthetic data (Section 4.2.5) makes sense. If a model of the data is built naively and the data contain outliers then sampling from

this model may reproduce some of the outliers [98]. de Waal and Willenborg [68] have shown that including sampling weights[1] in released data can increase re-identification risk if the sampling weights are computed from attributes (like geographical information) that are not released in the data because such attributes could be reconstructed with the help of sampling weights.

An interesting re-identification study was performed by Blien et al. [36] which studied the security of data in the face of a typical linkage attack. The data in this study came from a microcensus conducted in North Rhine-Westphalia in Germany in which 1% of the population was surveyed. Blien et al. used a handbook of German scientists as external file for the record linkage. They noted that linkage was affected by various errors and ambiguities in the data. This included data entry, attributes values that have changed over time, and multiple possible values for an attribute (such as "primary" occupation). Overall, if an attacker does not know that particular individuals are necessarily in the data, Blien et al. concluded that re-identification rate was not very high and many false positives (incorrect linkages) were found. They reasoned that the combination of data errors and subsampling was enough to thwart this particular attack.

While record linkage techniques generally try to link structured records to the data, it may also be possible to link free text. Novak et al. [193] show that it is possible to link posts on a message board to identify aliases. Text can also be linked to structured records: Frankowski et al. [105] link posts in a MovieLens forum to the MovieLens rating data set. The linkage can extend beyond simply matching based on the name of the movie since opinions can also be extracted from text [66, 197].

### 6.2.2 Composition

Sometimes, a privacy breach can occur when the external data themselves are also sanitized. Ganta et al. named this a *composition attack* [108]. This attack is possible when several data publishers own data sets

[1] These are weights associated with a tuple that tries to counteract selection bias by measuring how likely an individual is to respond to a survey and therefore appears in the data.

Table 6.5. Sanitized data released by two independent parties.

(a) Gotham Hospital's sanitized data

| Gender | Age | Zip | Disease |
|---|---|---|---|
| F | [21–35] | 10010 | Cancer |
| F | [21–35] | 10010 | Flu |
| F | [21–35] | 10010 | Allergy |
| F | [21–35] | 10010 | Malaria |
| M | [40–60] | 10010 | HIV |
| M | [40–60] | 10010 | Allergy |
| M | [40–60] | 10010 | Allergy |
| M | [40–60] | 10010 | Flu |
| M | [21–35] | 10024 | Scurvy |
| M | [21–35] | 10024 | Flu |
| M | [21–35] | 10024 | Varicella |
| M | [21–35] | 10024 | HIV |

(b) Gotbacon Hospital's sanitized data

| Gender | Age | Zip | Disease |
|---|---|---|---|
| F | [10–72] | 10010 | Allergy |
| F | [10–72] | 10010 | Flu |
| F | [10–72] | 10010 | Cancer |
| F | [10–72] | 10010 | HIV |
| F | [10–72] | 10010 | Flu |
| F | [10–72] | 10010 | Allergy |
| M | [11–60] | 10024 | Scurvy |
| M | [11–60] | 10024 | Allergy |
| M | [11–60] | 10024 | Cancer |
| M | [11–60] | 10024 | HIV |
| M | [11–60] | 10024 | Allergy |
| M | [11–60] | 10024 | Allergy |

that are not disjoint, but they still publish sanitized versions of their data independently of each other. This can happen, for example, for two hospitals in the same city. It would not be uncommon for patients to have been in both hospitals.

An example of a composition attack is shown in Tables 6.5(a) and 6.5(b). Gotham Hospital and Gotbacon Hospital, both in the same city, independently release sanitized versions of their data. An attacker might know that Bob just finished his Master's degree, is living in zip code 10024, and has been a patient at both hospitals for a recurring condition. An attacker may reason that Bob's age is likely within the 21–35 range in Gotham's data set and 11–60 range in Gotbacon's data set. By joining on the disease attribute, the attacker will see that the only diseases associated with Bob's demographic in both data sets are scurvy and HIV infection. Thus the attacker obtains a sharper inference about Bob's medical condition. This attack generalizes an observation of Sweeney's [241] which warns against publishing two different $k$-anonymous versions of the same data set. Similar problems have also been noted in the release of sanitized data over time [45, 131, 201, 260, 275].

### 6.2.3 Attacks using similar data

In many cases, it is possible to attack one sanitized data set with the help of a second data set even if they are disjoint. Lakshmanan

et al. [147] consider the following scenario, in which similar data can be a problem. Suppose a company sells items $I_i, \ldots, I_m$ and maintains a database which is a list $\{T_1, \ldots, T_n\}$ of transactions. Each transaction is the set of items purchased by the same customer at the same time. The company applies a perfect hash function (i.e., there are no collisions) to each item in each transaction in the data set and publicly releases the result. For each item $I_1, \ldots I_m$, the attacker has a belief function which gives upper and lower bounds on the frequency of an item in the company's data set (the belief function may have mistakes). This belief function can be formed from similar data (for example, by a competing company). The goal of the attacker is to match each item to its hashed value. Every correct match is a crack. Lakshmanan et al. [147] show that computing the expected number of cracks has a #P-complete subproblem and explore algorithms for generating approximate answers.

While Lakshmanan et al. [147] focus on frequency computation to break the anonymization, it is possible to use co-occurrence information as well. Malin [169] investigated the case when Websites share IP addresses of their users and also separately share the identities of their users. Co-occurrence information (which users went to a Website, and which IP addresses were recorded at a Website) can be used to match names with IP addresses.

Kumar et al. [146] investigated how co-occurrence information can reverse the anonymization of search query logs. The anonymization technique they studied was similar to [147]. Each word in each query was hashed with the same perfect hash function so that the final data set was a collection of multisets of integers. Since an attacker can get access to different, un-encoded search logs (such as the infamous AOL data set described in Section 1.2), it is important to make sure that the attacker cannot use such data to reverse the hash function. Kumar et al. [146] demonstrated the following negative result: frequency and co-occurrence information from the un-encoded search logs can be used to recover many of the hashed tokens from the hashed data set. The privacy risk comes from tokens that are frequent individually but infrequent in combination. For example, tokens such as "Cage" and "Denzel" may be easily recoverable from the names of famous actors, and if a person named "Denzel Cage" searched for his name, this query

would also be identified. The unsuitability of hashing as a primary means of anonymization has been demonstrated even further in the context of social networks. If a social network is published as simply a graph with no other information (such as attributes of a node) then re-identification attacks are still possible. Backstrom et al. [26] showed that if an attacker (or set of attackers) participates in the social network, then in many cases it is possible for the attacker to identify nodes corresponding to accounts under his control. We will describe anonymization techniques for social networks, search logs, and other emerging applications in more detail in Section 7.

### 6.2.4 Instance Level Background Knowledge

Finally, as described earlier in Section 2.4, it is often important to consider the role of instance-level background knowledge, or the type of knowledge that might be available to an acquaintance. In the context of $k$-anonymity, logical instance-level background knowledge has been studied by Machanavajjhala et al. [166, 168], Martin et al. [173], and Chen et al. [51]. (For more details, refer to the extended discussion in Section 2.4.)

While these ideas were based on logic, Du et al. [82] showed how to incorporate statistical knowledge. The statistical knowledge comes in the form of linear constraints and linear inequalities on probabilities. The inference of an attacker is modeled using maximum entropy. When background knowledge is expressed in the form of linear constraints, the use of maximum entropy has theoretical justifications [199]. When linear inequalities are introduced, the picture is not as clear. To see this, suppose we have a biased coin and we believe $P(\text{heads}) \geq 0.5$. If we use maximum entropy, we will assign $P(\text{heads}) = 0.5$. A reasonable alternative would posit that the bias of the coin is uniformly distributed in the interval $[0.5, 1]$ leading to the assignment $P(\text{heads}) = 0.75$. The choice, of course, depends on the attacker. To simulate and quantify the amount of knowledge an attacker may have, Du et al. [82] propose to mine the original data for positive and negative association rules. Freitas and Kuck [67] present a different approach to learning from population statistics based on a sparse probabilistic model. Li and Li

[158] propose to mine negative association rules and then use them to guide the anonymization processes. Ramesh [208] proposes to use samples to mine background knowledge and evaluates what can be learned from samples. Johnsten and Raghavan [135] propose to mine classification rules to evaluate the security of suppressing sensitive values, while Aggarwal et al. [9] present a defense against this. An additional reason for mining the original data is that an attacker may have knowledge about a small subset of these data. Xiong et al. [276] show how to use ideas from semi-supervised learning to make inferences from a sanitized data set if a subset of the original data are available.

# 7

# Challenges and Emerging Applications

> It may well be doubted whether human ingenuity can construct an enigma ... which human ingenuity may not, by proper application, resolve.
>
> — Edgar Allan Poe

The problems of privacy preservation, re-identification, and inference control are not limited to non-aggregate microdata and contingency tables. In an increasingly data-driven society, these issues are becoming important in a wide variety of emerging applications, where personal data are collected automatically. In many of these new applications, the privacy goal is generally *de-identification*, that is, the removal of personally identifiable information.

For example, in 2006 AOL made headlines when it released the purportedly de-identified search logs of many of its users. While the users' names were removed, many were easily identified based on the contents of their searches [28]. Similarly, it has been shown that individuals may be identified simply by the structure of social network graphs [26].

While the nature of the problems are similar, existing privacy criteria and definitions (e.g., $k$-anonymity, differential privacy, etc.) do not

necessarily directly apply to the privacy problems in these emerging domains, nor do existing sanitization mechanisms. This is still the subject of ongoing work in the research community, but in this section we give a series of example problems. Then, in Section 7.4, we summarize challenges for future research.

## 7.1 Social Network Privacy

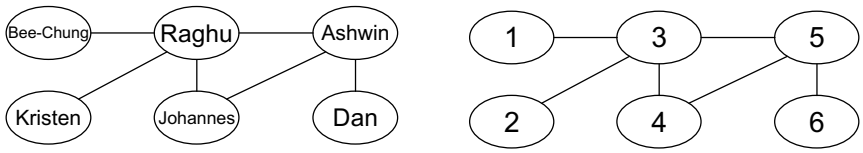
Social networks describe entities (often people) and the relationships between them. Social network analysis is often used to understand the nature of these relationships, such as patterns of influence in communities, or to detect collusion and fraud.

Collections of social network data have become pervasive on the World Wide Web. For example, e-mail messages implicitly define relationships between people. Social networking sites (e.g., Facebook and MySpace) and instant-messaging programs allow users to explicitly define such "friend" or "buddy" relationships. Making such data available can be invaluable to researchers, who seek to understand the dynamics of these communities. However, the release of data is often prevented by concerns about the privacy of individuals. In this section, we give an overview of various attacks that have been used to reveal private information from social network data, as well as counter-measures that have been proposed to reduce the risk of such attacks.

### 7.1.1 Naive De-Identification and Attacks

We will model a social network as a simple, undirected graph $G = (V, E)$. Nodes correspond to entities and edges represent connections between entities. Each entity has an associated unique name (e.g., Raghu or Johannes).

In designing a privacy-preserving publication scheme, the goal is to remove information pertaining to individual identities, while retaining the topological structure of the graph. To do this, one might consider a naive de-identification approach, whereby each node's name is replaced with a meaningless unique value (a *pseudonym*). For example, consider the social network in Figure 7.1(a). The naive solution would replace



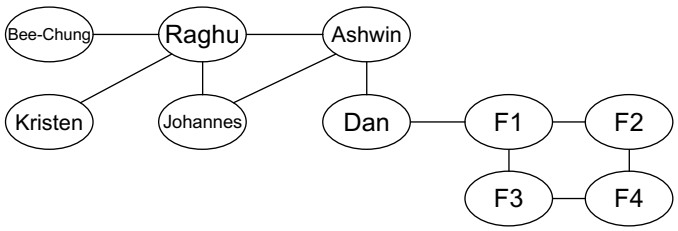

(a) Original data (b) De-identified data

Fig. 7.1 Social network example.

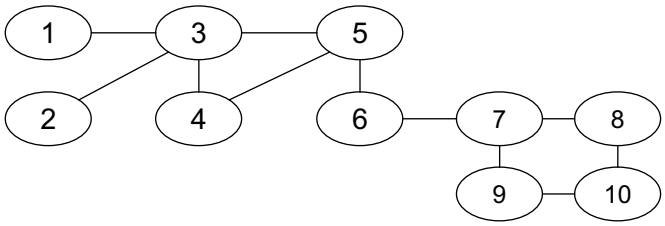
this network with the graph in Figure 7.1(b). We will refer to this de-identified graph as $G' = (V', E')$.

Unfortunately, there are various ways in which this naive solution can be compromised. Backstrom et al. [26] described two such attacks, one of which is *active* and the other is *passive*. In the family of *active* attacks, an attacker actively manipulates the structure of the graph before the data are released. For example, in online social networks, an attacker is able to create additional user accounts (corresponding to new nodes), and to create social connections among these accounts (new edges). These new nodes and edges form a subgraph $H$ of the social network $G$ and can make it difficult for a data publisher to release a truly de-identified version of $G$. For example, suppose Dan creates the fake accounts $F_1, \ldots, F_4$ and attaches them to the original network graph as shown in Figure 7.2(a). Provided that $H$ is sufficiently unique, the attacker may be able to locate $H$ in the naively de-identified graph $G'$, and then use this information to re-identify additional nodes that are connected to $H$ in the original graph $G$. For example, in Figure 7.2(b), once Dan locates the structure $H$, he is able to identify the nodes corresponding to himself and to Ashwin.

In contrast to active attacks, the family of *passive* attacks does not require any manipulation of the graph structure. Instead, these attacks can be launched based on background knowledge related to the graph's structure. It has been observed that many nodes belong to small, uniquely identifiable subgraphs [26, 123, 187]. Thus, with minimal background knowledge about the surrounding graph structure, it is often easy for an attacker to locate a target node in $G'$. For example, suppose Raghu knows that he has four connections in the original graph; he is able to locate himself easily in the de-identified graph shown




(a) Original Data




(b) De-identified data

Fig. 7.2 Active attack example.

in Figure 7.1(b). Now, suppose that Raghu would like to locate Dan. Simply knowing that Dan has one neighbor is not sufficient; however, if he learns that Dan's neighbor (Ashwin) has three connections in the graph, then he is able to locate Dan.

### 7.1.2 Prevention Techniques

While many questions remain, several techniques have been proposed recently, each with the goal of preventing attacks on published social network data. The proposed solutions differ in terms of the assumed threat models (specifically, the quantity information available to the attacker) and also in terms of the mechanisms used to prevent attack.

In one such work, Hay et al. presented an interesting framework for reasoning about (and preventing) passive structural re-identification [123]. The framework is based on the two key ideas of a *knowledge query* and *candidate set*. When attempting to locate a target individual $t$ in $G'$, an attacker may use various forms of structural background

knowledge. This is nicely abstracted in terms of a generic knowledge query $Q(t)$, which can also be evaluated over nodes in $G'$. If the attacker knows the answer to $Q(t)$, he can narrow the set of nodes that could possibly be $t$ to a candidate set $cand(t) = \{t' \in V' | Q(t) = Q(t')\}$.

In one of the simplest cases, we might consider knowledge queries that return the degree of node $t$. For example, an attacker may know that $deg(Ashwin) = 3$. Thus, looking at the de-identified graph in Figure 7.1(b) we have $cand(Ashwin) = \{5\}$.

Of course, the idea of knowledge queries is general enough to express far more than degree information. A significantly stronger form of knowledge is formalized through the idea of *strong structural knowledge* as follows. Two nodes $v_1, v_2 \in V'$ are said to be *automorphically equivalent* if there exists an isomorphism from $G'$ onto itself that maps $v_1$ to $v_2$. Intuitively, this corresponds to a strong knowledge query $Q(t)$ that returns the entire graph topology surrounding $t$. In this case, the candidate set of $t$ is restricted to the set of nodes in $V'$ to which $t$ is automorphically equivalent. For example, in Figure 7.1(b), nodes 1 and 2 are automorphically equivalent. Suppose that an attacker wants to locate Bee-Chung in this graph. Even if the attacker knows the entire graph topology surrounding Bee-Chung (i.e., the attacker has strong structural knowledge), he is still unable to determine whether node 1 or 2 represents Bee-Chung in $G'$.

Using this general framework, Hay et al. [123] formulated a logical privacy requirement which, given a particular language for expressing knowledge queries $Q$, requires that all candidate sets be of a required minimum size (i.e., $\forall t, |cand(t)| \geq k$, where $k$ is an input parameter) They also proposed a mechanism for satisfying this privacy requirement under strong structural knowledge. The mechanism is based on clustering. Informally, the idea is to partition the graph $G'$ into groups containing at least $k$ nodes each, and then to replace each group with a summary. Because of the strong adversarial model, this approach is sufficient to prevent a large class of passive re-identification attacks, including the attacks described by Narayanan and Shmatikov [187].

While the framework described by Hay et al. provides a semantic definition of privacy, including a precise characterization of background knowledge, the only information protected under this definition is the

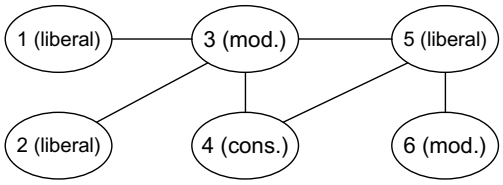



Fig. 7.3 De-Identified social network with sensitive attribute.

association between pseudonyms and true node identities. In some cases, this may not be enough. For example, even though an attacker does not know whether node 1 or 2 in Figure 7.1(b) corresponds to Bee-Chung, both of these nodes have precisely the same neighbors. Thus, if the information we truly mean to hide is not the association between names and pseudonyms, but instead some information about Bee-Chung's position or connections in the graph, then it does not matter whether the attacker believes Bee-Chung is node 1 or 2 because, in either case, the surrounding structure is the same.

Similarly, this framework does not explicitly provide support for limiting attribute disclosure, although we observe that the extension is straightforward. For example, consider a de-identified social network that includes not only a graph topology, but also certain attributes (e.g., Political Ideology) describing the nodes (see Figure 7.3). An attacker may be unsure whether node 1 or 2 is Bee-Chung, but in either case, the political ideology is "liberal." In much the same way as $\ell$-diversity, this problem can be addressed by requiring sufficient diversity of attribute values within each *automorphic equivalence class* (set of nodes that are automorphically equivalent to one another).

Other work has also considered preventing identity disclosure in social network graphs, but for less-powerful adversaries. Zhou and Pei [285] assume that an adversary is able to isolate a node $v$ only amongst the set of nodes in $G'$ sharing the same neighborhood structure; a neighborhood consists of a node's immediate neighbors and the connections between them. Liu and Terzi [163] assume that an adversary is able to isolate a node $v$ only amongst the set of nodes in $G'$ having the same degree. In contrast to Hay et al., these works propose mechanisms based on adding edges [285] or adding and deleting edges [163].

Of course, node re-identification based on graph structure is not the only threat to social network data. Recent work by Zheleva and Getoor [284] observed that, for real social network data, in which each node carries some associated attribute information, it is often possible to accurately predict missing (hidden for privacy purposes) attribute values. Other recent work by the same authors identifies the problem of link re-identification in published social networks [283].

## 7.2 Search Log Privacy

On July 29, 2006, AOL published three-month Web search queries of around 600 thousand users. For a given user, this data set contained the queries submitted by the user to the AOL search engine. To protect users' privacy, AOL replaced the AOL user names with randomly generated ID numbers. However, soon after the data set was released, many users together with their private queries were identified. As an example, the New York Times identified user No. 4417749 because this user searched for her family name, her hometown, and something about her age [28]. By combining this information, it was not difficult to create a very short list of candidates that matched the information. Several Websites even provided tools for anyone to look at the released search log, find user identities, and make comments about those AOL users. Several lawsuits were filed against AOL, and nine days after the release, AOL made an apology and terminated several employees involved in the data release, including the CTO.

The AOL case signifies the need for appropriate search log anonymization. Existing privacy definitions do not apply directly to search logs. In 2007 and 2008, several research studies analyzed privacy issues in search log publishing. However, to our knowledge, satisfactory solutions to search log publishing are still yet to be found. While Poblete et al. [202] believe that "... query log anonymization does not look promising in the near future, especially from the user privacy perspective ...," it is nonetheless an important direction for further research with some interesting on-going work. In the rest of this section, we first discuss the challenges of search log anonymization in Section 7.2.1, and then review a number of interesting proposals in Section 7.2.2.

### 7.2.1 Challenges

Table 7.1 shows an example search log of a search engine. Each time when a user submits a query or clicks a URL on the search result page, the user ID, the query, the clicked URL, if any, and some context information (e.g., time and the position of the URL) are logged by the Web server. This kind of search log is very useful for search engines to improve their search services and for the research community to advance the state-of-the-art in Web search. However, without appropriate sanitization, releasing such data could allow an attacker to identify individuals in the data and reveal their private queries. Possible attacks include:

- Obtaining sensitive information directly from queries: Some people (like user1) enter their social security numbers, credit card numbers, etc., in the search box.
- Identifying users by demographic attributes: Some people (like user2) search for information about their hometown, their age, and even their names. Note that, in the example search log, Arlington, WI, is a small town with less than 1,000 residents. It would not be difficult for an attacker to find the identity of user2.
- Identifying users by following URLs: Sometimes, the Web page that a user (like user3) searched for and clicked on can reveal the user's identity when combined with his/her other queries. For example, user3 is likely to be a pastor of a church,

Table 7.1. Example search log.

| | User ID | Query | Time | Rank | URL |
|---|---|---|---|---|---|
| 1 | User1 | Tax ssn 111223333 | 2008-01-05 08:10 | | |
| 2 | User2 | Restaurant arlington wi | 2008-01-03 10:20 | 1 | local.yahoo.com/... |
| 3 | User2 | Restaurant arlington wi | 2008-01-03 10:22 | 4 | www.gorestaurants.net/... |
| 4 | User2 | 70 single men | 2008-01-05 14:30 | | |
| 5 | User2 | chen family tree | 2008-01-06 20:01 | 1 | chenfamilytree.com |
| 6 | User2 | Nude pictures | 2008-01-10 21:42 | | |
| 7 | User3 | www.some-church.com | 2008-01-08 10:35 | 1 | www.some-church.com |
| 8 | User3 | Tax for pastor | 2008-01-13 22:50 | 8 | answers.yahoo.com/... |

[a]Rank and URL indicate the position and the URL on the search result page that the user clicked. Empty means no click.

and his/her name is likely to be found on the church's Website www.some-church.com.

- Discovering users' private queries: After identifying a user, an adversary can easily discover the user's private queries (e.g., "nude pictures" by user2) by looking at the entire search history of the user.

We note that other attacks are possible. The above list just provides a few examples showing potential threats. The challenges in search log anonymization include the following:

**No well-defined identifying or sensitive attributes:** Any search term (or phrase) used by a user is potentially sensitive, and the sensitivity depends on the semantics of the search term and context. Even if we can define and remove sensitive terms, they sometimes can still be predicted. Jones et al. [136] showed that, even if we are able to completely remove age, gender, zip code from a search log, the age, gender and zip code of a user can still be predicted based on his/her other queries. Another negative result obtained by Kumar et al. [146] shows that, even if we replace every search term with a secure hash number (so that the hash numbers do not reveal any information about the search terms), some frequent terms can still be identified based on co-occurrence analysis using another previously released search log (e.g., the AOL search log). They also showed that combinations of such terms can reveal private information about a number of users. Adar [5] proposed to sanitize a search log by removing queries with frequencies less than or equal to $k$. However, no formal result shows that removing infrequent queries guarantees privacy. For example, consider a search log in which all users searched for local information about a town also searched for pornography. Even if the number of such queries is large, by knowing a user lives in that town, we can still say with high confidence that he searched for pornography.

**Uniqueness of user searches:** A user search history is the set of all the queries submitted by the user. User search histories are almost all unique. Even a small set of queries from a user's search history is still likely to be unique, especially if the set contains less popular terms. To make a user search history less unique (or less identifiable),

anonymization techniques for set-valued data [246] may be used to generalize individual queries (or terms) in a user search history along some query hierarchy. However, it is not clear how one can build a query hierarchy and how useful the generalized queries would be. Adar [5] also proposed an approach for splitting a user search history into sessions (non-overlapping subsets of the history that cover the entire history) and only reveal sessions, not user search histories. He studied splitting by time of a query and splitting by text similarity (so that, for a given user, similar queries in terms of their text content are put in a session, while dissimilar queries are put in different sessions) and empirically showed that splitting by text similarity is "safer" than splitting by time. However, no formal privacy guarantee was provided.

**Possible adversarial manipulation:** An adversary can create multiple accounts and generate many queries using those accounts to create special query patterns, so that, when the search log is sanitized and released, the adversary can use those patterns to obtain private information about other users. For example, an adversary can generate a lot of queries to make an infrequent query frequent, or generate distinctive signatures to allow him/her to break the sanitization mechanism. To our knowledge, no research has addressed these issues in search log publishing. In related work, Backstrom et al. [26] studied a family of such attacks in the context of social network anonymization.

Finally, we note that, in addition to user privacy, Poblete et al. [202] studied Website privacy. Instead of users, they seek to protect the privacy of Websites whose URLs appear in the search log. For example, a Website may not want its competitors to know the terms used by users when searching for the Website. Several methods were proposed. However, no formal privacy guarantee was provided.

### 7.2.2 Interesting Proposals

We now review a number of interesting proposals for search log anonymization that try to formally define privacy criteria for user search queries. For a survey of practical techniques (i.e., deleting or hashing queries, deleting or hashing identifiers, removing known identifiers, deleting infrequent queries, shortening sessions) from the perspective of search company data retention policies, see [55].

**Differential privacy on search queries:** In order to give search log privacy a formal definition, Korolova et al. [144] apply a variant of differential privacy (discussed in Section 2.6) to search queries and develop an anonymization algorithm based on "noisy counts." For any statistic $x$ of a search log (e.g., number of times a given query appears in the search log), the noisy count of $x$ is $x + \epsilon$, where $\epsilon$ is random noise drawn independently from the Laplace distribution with mean zero and scale parameter $b$, for some carefully chosen $b$. Intuitively, their algorithm works as follows.

(1) **Limit user activities:** For each user, retain only the first $d$ queries by the user and the first $d_c$ URL clicks.
(2) **Process queries:** For each query, add noise to the number of times the query appears in the search log. If this noisy count exceeds a specified threshold, output the query with its noisy counts.
(3) **Process URLs:** For each query that has a noisy count exceeding the threshold, output the noisy count of the number of times each URL was clicked for that query.

The output from the algorithm consists of frequent queries together with noisy counts of the queries and clicked URLs. Korolova et al. showed that the sanitized search log can be used for applications such as keyword generation and studying people's fears (using queries containing the words "fear of") with reasonable performance. However, they also note that whether it can be useful for other applications is "far from being answered and is an interesting avenue for future work."

**Plausibly deniable search:** Murugesan and Clifton [184] address search query privacy by a client-side approach and propose the notion of plausibly deniable privacy. When a user wants to submit a query $q^*$ to a search engine, a client-side tool generates a set $Q = q_1, \ldots, q_k$ of "cover" queries and submit all the cover queries to the search engine instead of $q^*$ ($q^*$ may or may not in $Q$). The idea is that there must exist a query $q_i \in Q$ that is very similar to (or the same as) $q^*$, so that the user can get the desired search results (by filtering out results returned from the search engine for queries $q_j$, for $j \neq i$). Also, the set $Q$ must be sufficiently diverse, the probability that the client-side

tool generates $Q$ from any one of $\{q^*, q_1, \ldots, q_k\}$ should be the same, and each query $q^*, q_1, \ldots, q_k$ should have equal probability of being the user's actual query, so that the user can deny his actual query $q^*$ (or $q_i$). Murugesan and Clifton then define similarity and diversity using methods (cosine similarity and latent semantic indexing) developed in the information retrieval (IR) literature, and provide an algorithm to generate a desirable set $Q$ of cover queries. This work is an interesting step toward client-side privacy preservation of search queries. However, the exact privacy guarantee is defined based on the performance of the IR techniques used, which may not be desirable. Also, how useful the sanitized data are and how to protect privacy for query sequences are questions that need further research.

**Sketch-based anonymization:** Finally, we discuss the work by Aggarwal and Yu [11] on text data anonymization that can also be applied to search logs. The idea is to replace a user's search history by a set of sketches [21]. The goal is to prevent an attacker from identifying the query terms used by an individual but to still allow a researcher to compute similarities between the sanitized queries (for example, by using cosine similarity) with reasonable accuracy.

Suppose each user's search history is represented by a bag of terms. Let $x_u = [x_{u1}, \ldots, x_{ud}]$ denote the search history of user $u$, where $x_{ut}$ is the number of times that term $t$ occurs in $u$'s search history, and $d$ is the total number of terms in the entire search log. Note that we enumerate all the terms in the entire search log and represent each term as its sequence number. This term-to-number mapping is public. Also note that metadata like time, rank, and URL is lost, and that $x_u$ is a sparse vector in which most of the elements are zero. Our goal is to sanitize $x_u$ by replacing it with a vector of sketch components $s_u = [s_u^1, \ldots, s_u^{m(u)}]$, where each $s_u^j$ is a number and $m(u)$ is the number of sketch components for user $u$. The number $m(u)$ controls the amount of privacy in the sanitized data and needs to be chosen according to a privacy criterion such as $\delta$-anonymity or $k$-variance (which are discussed below). Note that each user may have a different number of sketch components. Let $M$ denote the largest number of sketch components considered by the algorithm.

To sanitize the search log, we first generate $M$ sequences of 4-wise independent pseudo-random numbers, where each sequence $r^j = [r_1^j, \ldots, r_d^j]$ has length $d$ and each $r_t^j$ is either 1 or $-1$. Then, for each user $u$, we sanitize $x_u$ by replacing it with $s_u$ where

$$s_u^j = \sum_{t=1}^{d} x_{ut} \cdot r_t^j.$$

It can be easily shown that $\hat{x}_{ut}(s_u)$, defined as follows, is an unbiased estimator of $x_{ut}$.

$$\hat{x}_{ut}(s_u) = \frac{1}{m(u)} \sum_{j=1}^{m(u)} s_u^j \cdot r_t^j,$$

with mean $E[\hat{x}_{ut}(s_u)] = x_{ut}$ and variance

$$Var[\hat{x}_{ut}(s_u)] = \frac{1}{m(u)} \left[ \left( \sum_{k=1}^{d} x_{uk}^2 \right) - x_{ut}^2 \right].$$

Note that the more sketch components ($m(u)$) we have, the more accurately we can recover the original data ($x_{ut}$), because the variance of the estimator $Var[\hat{x}_{ut}(s_u)]$ reduces as $m(u)$ increases.

The inner product $x_u \cdot x_v$ is a commonly used measure of similarity between two bags of words $x_u$ and $x_v$. An unbiased estimator for this inner product is $sim(s_u, s_v)$, defined as follows:

$$sim(s_u, s_v) = \frac{1}{\rho} \sum_{j=1}^{r} s_u^j \cdot s_v^j,$$

where $\rho = \min\{m(u), m(v)\}$. It has mean $E[sim(s_u, s_v)] = x_u \cdot x_v$ and variance

$$Var[sim(s_u, s_v)] = \frac{1}{\rho} \left[ \|x_u\|^2 \|x_v\|^2 - (x_u \cdot x_v)^2 \right]$$

(where $\|x_u\|^2 = \sum_{t=1}^{d} x_{ut}^2$). Here $\rho$ controls the precision of the estimator.

Two privacy criteria associated with this technique are $\delta$-anonymity and $k$-variance (defined below). Assume $m(u)$ is given for each user $u$. Let $D = \{x_u\}$ denote the original search log and $D^* = \{s_u\}$ denote the sanitized version of the search log that results from this sketch-based technique.

**Definition 7.1 ($\delta$-Anonymity).** The release candidate $D^*$ is $\delta$-anonymous if $Var[\hat{x}_{ut}(s_u)] > \delta$, for every user $u$ and term $t$.

$\delta$-Anonymity ensures that the uncertainty in the reconstructed value $\hat{x}_{ut}$ of each term frequency component $x_{ut}$ is at least $\delta$. As noted by Aggarwal and Yu, a disadvantage of $\delta$-anonymity is that it treats each user independently regardless of whether there are other users similar to him/her. They argued that it is desirable to give outliers (users who use unique terms) more protection than users who are similar to many others. Thus, they define the $k$-variance criterion.

**Definition 7.2 ($k$-Variance).** Release candidate $D^*$ satisfies $k$-variance if, for any two users $u$ and $v$ such that $x_u$ is among $x_v$'s $k$-nearest neighbors, $Var[sim(s_u, s_v)] \geq |x_u| \cdot |x_v| - |x_u \cdot x_v|$.

The aim of $k$-variance is to ensure that any user $u$'s sanitized search history cannot be easily distinguished from its $k$-nearest neighbors. Note that if we divide each vector $x_w$ by a constant $c$ then the variance of $sim(s_u, s_v)$ decreases by a factor of $c^4$ while $|x_u| \cdot |x_v| - |x_u \cdot x_v|$ decreases by a factor of $c^2$. Thus normalization of the data is extremely important. Aggarwal and Yu recommend rescaling each $x_w$ so that $|x_w| = 1$.

Aggarwal and Yu described algorithms for $\delta$-anonymity and $k$-variance in [11] and noted that some of the vectors $x_u$ may need to be suppressed in order to achieve $\delta$-anonymity or $k$-variance. It is interesting to see how useful the sketch-based method would be when applied to real search logs, whether or not the attacks described in Section 7.2.1 would succeed, and what kinds of search log analysis can still be conducted with acceptable accuracy when we only have sanitized search logs. One should also be careful of releasing the (pseudo)randomly generated $r^j$ that were used in the sanitization process since this may allow linear programming techniques to reconstruct the original data [74, 88].

## 7.3 Location Privacy and Mobile Applications

Privacy protection is also increasingly important in emerging mobile and location-aware applications. For example, many modern cellular phones and cars now come equipped with global positioning system (GPS) devices. Thus, cellular service providers and car companies are able to collect location trace data from many mobile users. The owners of these repositories may wish to publish, distribute, or sell these data to enable a new set of applications called *location-based services* (LBS), including highway traffic and safety analysis, and strategic placement of outdoor advertisements (e.g., billboards).

At an intuitive level, the nature of these applications suggests a potential threat to individual privacy. However, the privacy threat can vary by application, and is not always crisply stated, so it can be difficult to judge whether a particular protocol is successful in protecting personal privacy and sensitive information. In this section, we give an overview of the threat models (formal and informal) described in the literature, as well as an overview of privacy-protection mechanisms developed in response to these threats.

### 7.3.1 Spatial Cloaking for LBS

Early work in location privacy took much the same approach as work in anonymization for static microdata. Using location data from a single point in time, this work proposed replacing the locations of individual users with *cloaking regions*, such that each cloaking region contains at least $k - 1$ other users [109, 110, 119, 137, 180].

In the context of location-based services, several different system architectures have been proposed [112, 180], each allowing users to replace their specific locations with $k$-anonymous cloaking regions when making location-based requests. For example, consider a user named Jennifer who would like to use her cellular phone to find the organic coffee shop closest to her current location in Redmond. Instead of providing her exact location to the service provider, her location can be replaced with a region containing at least $k - 1$ other users. Of course, the service may not be able to answer the request precisely using the

cloaking region [180]. In this example, the service may instead return multiple coffee shops, and Jennifer's phone must filter the answers to find the one that is closest.

Informally, the cloaking approach is used in response to several, slightly different, threats. Consider requests of the form $(id, \ell, q)$, where $id$ uniquely identifies the user (e.g., Jennifer), $\ell$ is the user's location (e.g., a street corner in downtown Redmond), and $q$ is the content of the query (e.g., organic coffee shop). Using the cloaking approach, $\ell$ is often replaced with a $k$-anonymous cloaking region, $\ell^*$. In each of the following cases, the service provider is considered untrusted.

- *The identity of the user is considered sensitive.* This threat is most analogous to the traditional record linkage attack. In this case, the user removes her identity, and submits a request of the form $(-, \ell^*, q)$.[1] The adversary (malicious service provider) is assumed to have access to auxiliary information associating individuals with particular locations (e.g., telephone directories), so the location variable constitutes a quasi-identifier. Replacing the precise location with a $k$-anonymous cloaking region introduces uncertainty about the identity of the individual issuing the request.
- *The association between identity and query is considered sensitive.* This threat is analogous to the sensitive-value attack addressed by $\ell$-diversity [166]. Again, the user removes her identity, and submits a request of the form $(-, \ell^*, q)$, and the adversary is able to associate individuals with particular locations. However, in this case, rather than hiding her identity, the user wants to hide the content of her query (e.g., adult bookstores). Replacing the precise location with an $\ell$-diverse cloaking region introduces uncertainty about the association between identity and the value of the query.
- *The location of the user is considered sensitive.* Finally, suppose the user submits a request of the form $(id, \ell^*, q)$. The request includes the user's identity, but suppose that the user does not want to reveal his precise location. This threat is

[1] In location-based service systems, this typically requires the use of an anonymous routing protocol or trusted third party.

not specifically addressed by $k$-anonymous cloaking because the size of $\ell^*$ is dictated by the density of users. Rather, in this case, it is useful to allow the user to specify a minimum region size [180].

There are a variety of interesting extensions to these basic problems and threats. One that is particularly interesting is the set of unique problems posed by continuous queries. In this case, static (one-time) spatial cloaking may not be sufficient to protect individual anonymity, since users may be tracked using unique queries. Thus, additional precautions must be taken [32, 52].

Finally, spatial cloaking is just one way to address privacy in location-based services. For example, departing from the cloaking approach entirely, Ghinita et al. developed techniques based on private information retrieval (PIR) to hide the content of users' location-based queries from an untrusted service provider [111].

### 7.3.2 Anonymity for Location Traces and Trajectories

In addition to providing location-based services, mobile devices present an opportunity to collect, analyze, and distribute location data. This can be done *offline* (i.e., compile a database of trajectories to sell later) or *online* (i.e., continuously collect and distribute location information from a population of users). As an example of the former, a cellular phone company may wish to collect location trace information, and then sell this information to an advertising agency to assist in the placement of billboards. As an example of the latter, it may be valuable for a public-safety administration to monitor traffic patterns in real-time. Of course, location-tracking applications often raise privacy concerns among users, so considerable effort has focused on developing tools to help balance the goals of providing useful data while protecting individual privacy.

#### 7.3.2.1 Offline Anonymization for Trajectory Data

Recent work has developed variations of $k$-anonymous cloaking that can be applied (offline) to databases of fully-specified trajectories [3, 189, 245].

Consider, for example, the framework proposed by Abul et al. [3]. In this work, a trajectory $\tau$ is defined by a set of spatio-temporal points $(x_1, y_1, t_1), \ldots, (x_n, y_n, t_n)$ where $t_1 < \cdots < t_n$, as well as an additional parameter $\delta$, which describes the degree of uncertainty for each point in the trajectory. A set $S$ of trajectories is said to satisfy $(k, \delta)$-anonymity if (1) $|S| \geq k$ and (2) for each pair of trajectories $\tau_1, \tau_2 \in S$, the distance between each time-corresponding pair of locations is less than $\delta$ (i.e., for each time $t$ covered by the trajectories, $dist((x_1, y_1), (x_2, y_2)) \leq \delta$). For example, the two trajectories shown in Figure 7.4 satisfy this property.

Given this adapted privacy requirement, the challenge is to transform a database $D$ of trajectories into a database $D'$ such that for each $\tau \in D'$, there exists $S \subseteq D'$ such that $\tau \in S$ and $S$ satisfies $(k, \delta)$-anonymity. This should be done, ideally, in such a way that the transformation minimally affects the quality of the data. One proposed mechanism involves clustering the trajectories in $D$, and then cloaking (increasing the uncertainty) for the trajectories in each cluster [3].

#### 7.3.2.2 Online Anonymization for Trajectory Data

The online case presents a further challenge. In the offline case, a finite database of fully specified traces is compiled, anonymized, and

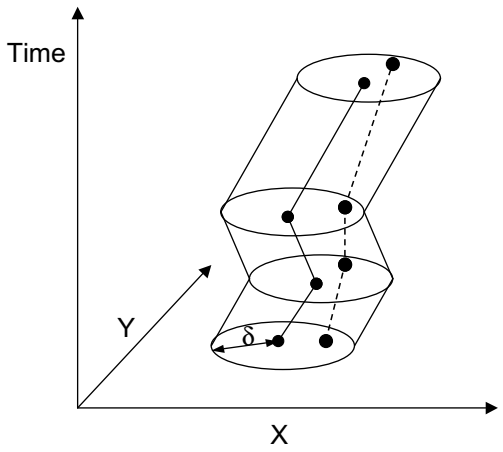



Fig. 7.4 Uncertainty-based trajectory cloaking.

released. In the online case, however, location trace information must be collected, anonymized, and distributed in (near) real-time. Existing offline anonymization tools do not directly apply because, after the initial anonymized trajectories are published, it is not clear how to safely publish future location information.

This is still very much an active area of research. In the remainder of this section, we seek to provide intuition for how online trajectory anonymization has evolved and developed over the past several years.

**Strawman 1:** Consider a finite population of $n$ users, each with a unique identifier in the set $(u_1, \ldots, u_n)$. For the sake of illustration, consider a strawman protocol that replaces each identifier with a unique pseudonym (e.g., a hash value) in the set $(p_1, \ldots, p_n)$. Unfortunately, this simple strawman is vulnerable to attacks based on *auxiliary information*.[2] In particular, we expect to encounter an adversary with access to some source of information that allows him to associate individuals with particular locations at particular points in time. (For example, an adversary might use the Yellow Pages to locate Jennifer's home, and then reason that she is likely to be at her home during the night.) Worse, in this simple strawman, once the adversary "unmasks" a user (e.g., determines that $p_3$ is Jennifer), he can learn Jennifer's location at other points in time.

**Strawman 2:** To overcome the shortcomings of the strawman, one might consider eliminating the use of time-consistent pseudonyms. For simplicity, assume that users' locations are reported in discrete time steps $t_0, t_1, \ldots$. In this second strawman approach, at each time $t_i$, we would replace user identifier $u_j$ with a pseudonym $p_j^i$ such that there is no discernible relationship between pseudonyms $p_j^i$ for $i = 0, 1, \ldots$.

Unfortunately, this approach has also been shown ineffective. Using multi-target tracking tools, several recent works have demonstrated that it is still often possible to track a particular user across time [120, 145]. Fundamentally, this attack stems from the fact that motion is (at least partially) predictable. As a simple example, consider a set of three users, as shown in Figure 7.5. Suppose that at 8:30 AM, the locations

[2] This can be equated with the idea of location as a *quasi-identifier* in the microdata anonymization and $k$-anonymity literature [32].

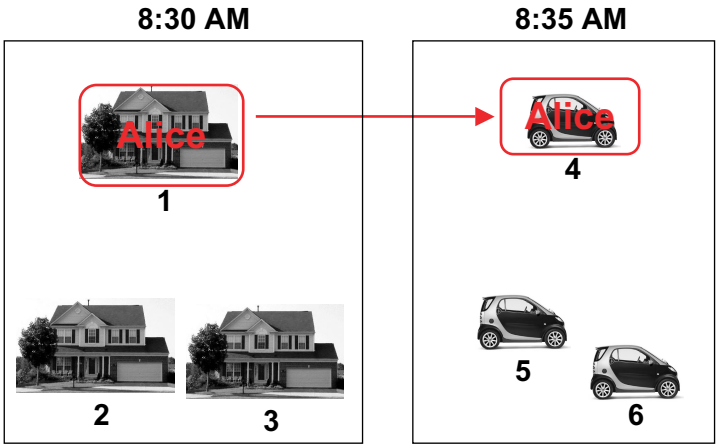



Fig. 7.5 Inference as the result of motion prediction.

of users (1, 2, 3) are published. Using auxiliary information, we already know that an adversary may be able to identify some of the users (in this case, Alice). Now, suppose that the users change locations, and at 8:35 AM, suppose that we publish the locations of users again (locations 4, 5, 6). By this time, Alice is on her way to work. However, it may still be possible to determine that Alice is at location 4 using a basic model of user motion. This is possible, for example, if the other locations (5 and 6) are too far away from location 1 to have been reached in the intervening five minutes.

**Temporal Unlinkability and Proposed Mechanisms:** Based on these observations about the expected threat model, the idea of using *temporal unlinkability* as a privacy requirement has developed. (This principle was formalized in [134], but the same intuition has also driven past work [31, 124].)

---

**Definition 7.3 (Temporal Unlinkability Principle [134]).** Suppose an adversary is able to correctly associate a user id $u_j$ with $m$ sequential pseudonyms, $p_j^i, \ldots, p_j^{i+m}$ during times $i, \ldots, i+m$. Under reasonable inference assumptions, the adversary should not be able to determine, with high confidence, the pseudonym $p_j^h$ corresponding to $u_j$ at some other point in time $h \notin \{i, \ldots, i+m\}$.

---

Various mechanisms have been proposed with the (formal or informal) goal of enforcing temporal unlinkability. Examples include mix-zones [31], location-sampling [124], and clustering [134]. Jin et al. [134] also present a formal framework for reasoning about temporal linkability in the presence of motion prediction.

## 7.4 Additional Challenges

While emerging applications of privacy provide their own set of challenges, there are some challenges that are application independent. In the remainder of this section, we will discuss some of these fundamental challenges.

### 7.4.1 The Curse of Dimensionality

One of the challenges in the area of privacy-preserving data publishing is caused by the ease with which data are collected. With improving technology it is becoming easier to measure and record more information about each individual. Thus, the number of attributes is growing, causing the size of the domain to increase exponentially. When this happens, the *curse of dimensionality* starts to cause information loss in sanitized data sets.

This effect can be explained by the following observations of Beyer et al. [33] about nearest neighbor queries. Under general circumstances, as dimensionality increases, the ratio of distances to nearest neighbor and farthest neighbor of a random query point approaches 1. This implies that in many cases, the distance between any two points in a fixed-size data set is very large. Thus for example, when grouping together data points for $k$-anonymity using local or global recoding, we will generally be putting very dissimilar points in the same partition, which would require many generalizations to be performed, causing much information loss. On the other hand, when adding random noise to the data, we will need to add quite a bit of noise to each data point to make it nearly indistinguishable from its nearest neighbors since those neighbors will generally be far away. Similar issues occur with synthetic data [167]. These results have been noted in theory [6, 7] as well as practice [39, 186], but have also been part of folklore for some time.

Some current approaches to mitigating the curse of dimensionality include using local recoding [150] or clustering [14] instead of global recoding, performing a lossy join decomposition between the quasi-identifiers and the sensitive attributes [273], publishing multiple views of the data [75, 142, 278] and publishing low-dimensional approximations of the data [113]. There has also been recent focus on anonymizing sparse high-dimensional data [11, 114]. This is a promising direction since the sparseness of the data indicates that it has a lower-dimensional signal, which may be used to help avoid the typical excessive information loss in high-dimensional data sanitization.

### 7.4.2 Sequential Releases and Composability

Privacy of sequential releases is another important piece in the privacy puzzle. The US Census Bureau publish data from the decennial census every 10 years; other data sets from the Census Bureau and related statistical agencies are published at more frequent (e.g., annual) intervals. Web applications collect data incrementally and would like to use the most current information; for instance, Netflix and Amazon frequently update their recommendations based on new transactions. These sequential releases pose an additional privacy threat since user information can be linked across different releases. For example, consider again the sanitized data shown in Table 2.2, which achieves 3-diversity through generalizations. An adversary cannot tell whether Bruce has cancer, heart disease, or the flu. Suppose another table is published by the same hospital and Bruce is in a 4-anonymous group, where two of the individuals have cancer, one has hepatitis, and one has stomach ulcer. Since Bruce has the same disease in both the tables, an adversary can deduce that Bruce has cancer despite he being in a 3-diverse group in both tables.

The main reason such an attack is possible is that even though $\ell$-diversity guarantees privacy against adversaries with background knowledge, it restricts adversarial background knowledge to only consist of $\ell - 2$ negation statements. However, in a sequential release, the adversary has access to prior releases; this knowledge cannot be captured in terms of $\ell - 2$ negation statements. Hence, definitions like $\ell$-diversity do not automatically guarantee the privacy of sequential

releases. Reasoning about posterior probabilities of sensitive attributes from data consisting of a set of generalized tables generally requires the construction of graphical models, which is NP-hard in general. Consequently, checking a privacy condition like $\ell$-diversity, $(c, k)$-safety, or privacy skyline would also be hard.

An alternate approach to handling sequential releases is using *composable* privacy definitions.

---

**Definition 7.4 (Composable Privacy Definition).** Let $P$ be a privacy definition that takes a parameter $\Theta$. For any release $R_i$, let $\Theta(R_i)$ be the value of the parameter $\Theta$ associated with $R_i$. Assume that any sequence of releases $\vec{R} = (R_1, R_2, \ldots, R_n)$ also satisfies $P$ with some privacy parameter $\Theta(\vec{R})$. We say that $P$ is $f$-composable if $\Theta(\vec{R}) = f(\Theta(R_1), \Theta(R_2), \ldots, \Theta(R_n))$ for any $\vec{R} = (R_1, R_2, \ldots, R_n)$.

---

Composability essentially means that the privacy guarantees degrade gracefully as more sanitized data sets, describing the same individuals, are released. Recently, Ganta et al. [108] initiated a formal study of composable privacy definitions. They showed that differential privacy is composable; i.e., if one release $R_1$ satisfies $\epsilon_1$-differential privacy and a second release $R_2$ satisfies $\epsilon_2$-differential privacy, then the combined sequential release satisfies $(\epsilon_1 + \epsilon_2)$-differential privacy. The above property is true even when the two releases are generated by different sanitization schemes. Ganta et al. also identify two relaxations of differential privacy, namely, $(\epsilon, \delta)$-probabilistic differential privacy [167] and $\delta$-approximate $\epsilon$-differential privacy [86], as composable privacy definitions. However, differential privacy is very strict, and identifying weaker composable privacy definitions is an important research direction. Additional perspectives on sequential releases are presented in [45, 131, 201, 260, 275], but note that data released by two different data publishers can also be a case of sequential release.

### 7.4.3 Obtaining Privacy Preferences and Setting Parameters

Though a system for sharing private data is useful, and recent proposals provide rigorous mathematical guarantees, there still exist usability

obstacles that, unless resolved, may stand in the way of widespread adoption.

One interesting usability question concerns expression of privacy preferences. For instance, consider a system like `delicious.com` that lets users tag Web pages, and allows them to share their tags with friends in an underlying social network. Suppose that Dan tags `http://www.abc.net` as `junk`. Currently, Dan can either choose to share this tag with Tony, in which case Tony knows for sure that Dan tagged the page as `junk`, or not share the tag, in which case Tony has no information about whether Dan tagged the page. However, notice that this sharing is binary, and Dan has no way of specifying that his tags should be shared anonymously with Tony. In order to anonymously share information, simple binary notions of sharing will not suffice. While technology supports sharing with provable guarantees of partial disclosure (e.g., $\ell$-diversity, differential privacy, etc.), we do not know how to elicit from users their preferences for allowable limits on disclosure in a way that the users understand, and about which they feel comfortable.

Another usability challenge arises from parameter setting. While mathematically rigorous privacy definitions have emerged, they are often fraught with user-specified parameters. In some cases, these parameters have interpretable meaning (e.g., the number of *pieces* of Boolean background knowledge necessary to breach privacy in the 3D privacy criterion, or the number of individuals whose information has to be held out to make a difference in $(c, \epsilon)$-differential privacy). However, in practice, it is still not necessarily clear how to set these parameters. One recent proposal suggested using game-theoretic information-sharing models as a means of reasoning about the availability of Boolean background knowledge to attackers, which in turn helps in parameter setting for privacy definitions based on quantities of background knowledge [84]. However, this problem is not solved in general and significant challenges remain.

# 8

# Conclusions

In an increasingly data-driven society, personal information is often collected and distributed with ease.

In this survey, we have presented an overview of recent technological advances in defining and protecting individual privacy and confidentiality in data publishing. In particular, we have focused on organizations, such as hospitals and government agencies, that compile large data sets, and must balance the privacy of individual participants with the greater good for which the aggregate data can be used.

While technology plays a critical role in privacy protection for personal data, it does not solve the problem in its entirety. In the future, technological advances must dovetail with public policy, government regulations, and developing social norms.

The research community has made great strides in recent years developing new semantic definitions of privacy, given various realistic characterizations of adversarial knowledge and reasoning. However, many challenges remain, and we believe that this will be an active and important research area for many years to come.

# Acknowledgments

We gratefully acknowledge the insight and support of our advisors and research collaborators, including Raghu Ramakrishnan, Johannes Gehrke, David DeWitt, Rakesh Agrawal, and John Abowd.

We would also like to thank Evimaria Terzi for discussions about the related literature in social network anonymization, Michaela Goetz for new insights on privacy definitions; Wen Jin and Jignesh Patel for discussions related to privacy for moving objects; and Bradley Malin for references and insights into work in genomic privacy.

# References


[1] J. M. Abowd and S. D. Woodcock, "Disclosure limitation in longitudinal linked data," *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, pp. 215–277, 2001.

[2] J. M. Abowd and S. D. Woodcock, "Multiply-imputing confidential characteristics and file links in longitudinal linked data," in *Privacy in Statistical Databases*, 2004.

[3] O. Abul, F. Bonchi, and M. Nanni, "Never walk along: Uncertainty for anonymity in moving objects databases," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2008.

[4] N. Adam and J. Wortmann, "Security-control methods for statistical databases," *ACM Computing Surveys*, vol. 21, no. 4, pp. 515–556, 1989.

[5] E. Adar, "User 4xxxxx9: Anonymizing query logs," in *Query Log Analysis Workshop at WWW*, 2007.

[6] C. C. Aggarwal, "On k-anonymity and the curse of dimensionality," in *Proceedings of the 31st International Conference on Very Large Databases (VLDB)*, 2005.

[7] C. C. Aggarwal, "On randomization, public information and the curse of dimensionality," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2007.

[8] C. C. Aggarwal, "On unifying privacy and uncertain data models," in *Proceedings of the 24th International Conference on Data Engineering (ICDE)*, pp. 386–395, 2008.

[9] C. C. Aggarwal, J. Pei, and B. Zhang, "On privacy preservation against adversarial data mining," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[10] C. C. Aggarwal and P. S. Yu, "A condensation approach to privacy-preserving data mining," in *Proceedings of the 9th International Conference on Extending Database Technology (EDBT)*, 2004.

[11] C. C. Aggarwal and P. S. Yu, "On privacy-preservation of text and sparse binary data with sketches," in *SDM*, 2007.

[12] C. C. Aggarwal and P. S. Yu, *Privacy-Preserving Data Mining: Models and Algorithms*. Springer, 2008.

[13] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu, "Approximation algorithms for k-anonymity," *Journal of Privacy Technology (JOPT)*, 2005.

[14] G. Aggarwal, T. Feder, K. Kenthapadi, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving anonymity via clustering in a metric space," in *Proceedings of the 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 2006.

[15] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proceedings of the 20th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 2001.

[16] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2000.

[17] R. Agrawal, R. Srikant, and D. Thomas, "Privacy Preserving OLAP," in *Proceedings of the 23th ACM SIGMOD Conference on Management of Data*, 2004.

[18] S. Agrawal and J. R. Haritsa, "A framework for high-accuracy privacy-preserving mining," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2005.

[19] D. J. Algranati and J. B. Kadane, "Extracting confidential information from public documents: The 2000 department of justice report on the federal use of the death penalty in the United States," *Journal of Official Statistics*, vol. 20, no. 1, pp. 97–113, 2004.

[20] P. Allison, "Multiple imputation for missing data: A cautionary tale," *Sociological Methods and Research*, vol. 28, no. 3, pp. 301–309, 2000.

[21] N. Alon, Y. Matias, and M. Szegedy, "The space complexity of approximating the frequency moments," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (STOC)*, 1996.

[22] S. Andrews and T. Hofmann, "Multiple-instance learning via disjunctive programming boosting," in *Advances in Neural Information Processing Systems 16 (NIPS)*, (S. Thrun, L. Saul, and B. Schölkopf, eds.), Cambridge, MA: MIT Press, 2004.

[23] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[24] M. Arrington, "Aol proudly releases massive amounts of private data," TechCrunch: http://www.techcrunch.com/2006/08/06/aol-proudly-releases-massive-amounts-of-user-search-data/, August 6, 2006.

[25] F. Bacchus, A. J. Grove, J. Y. Halpern, and D. Koller, "From statistics to beliefs," in *National Conference on Artificial Intelligence AAAI*, 1992.

[26] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, 2007.

[27] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar, "Privacy, accuracy and consistency too: A holistic solution to contingency table release," in *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 2007.

[28] M. Barbaro and T. Zeller, "A face is exposed for AOL Searcher no. 4417749," *New York Times*, August 9 2006.

[29] R. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymity," in *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, 2005.

[30] R. Benedetti and L. Franconi, "Statistical and technological solutions for controlled data dissemination," in *New Techniques and Technologies for Statistics*, 1998.

[31] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *Pervasive Computing*, vol. 2, no. 1, 2003.

[32] C. Bettini, X. Wang, and S. Jajodia, "Protecting privacy against location-based personal identification," in *Proceedings of the VLDB Workshop on Secure Data Management*, 2005.

[33] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "Nearest Neighbor" meaningful?," in *Proceedings of the 10th International Conference on Database Theory (ICDT)*, pp. 217–235, 1999.

[34] J. Bi and T. Zhang, "Support vector classification with input data uncertainty," in *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[35] Y. M. Bishop, S. E. Fienberg, and P. W. Holland, *Discrete Multivariate Analysis: Theory and Practice*. The MIT Press, 1977.

[36] U. Blien, H. Wirth, and M. Muller, "Disclosure risk for microdata stemming from official statistics," *Statistica Neerlandica*, vol. 46, no. 1, pp. 69–82, 1992.

[37] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: The SuLQ framework," in *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, 2005.

[38] G. E. P. Box and N. R. Draper, *Empirical Model-Building And Response Surface*. Wiley, 1987.

[39] J. Brickell and V. Shmatikov, "The cost of privacy: Destruction of data-mining utility in anonymized data publishing," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[40] S. Bu, L. V. Lakshmanan, R. T. Ng, and G. Ramesh, "Preservation Of patterns and input–output privacy," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2007.

[41] D. Burdick, P. M. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan, "Efficient allocation algorithms for OLAP over imprecise data," in *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, pp. 391–402, VLDB Endowment, 2006.

[42] D. Burdick, P. M. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan, "OLAP over uncertain and imprecise data," *VLDB Journal*, vol. 16, no. 1, pp. 123–144, 2007.

[43] D. Burdick, A. Doan, R. Ramakrishnan, and S. Vaithyanathan, "OLAP over imprecise data with domain constraints," in *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB)*, pp. 39–50, VLDB Endowment, 2007.

[44] J. Burridge, "Information preserving statistical obfuscation," *Statistics and Computing*, vol. 13, pp. 321–327, 2003.

[45] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure anonymization for incremental datasets," in *SIAM Conference on Data Mining (SDM)*, 2006.

[46] S. Canada, "The research data centres (RDC) program," http://www.statcan.gc.ca/rdc-cdr/network-reseau-eng.htm, June 9, 2009.

[47] G. Casella and R. L. Berger, *Statistical Inference*. Duxbury, 2nd ed., 2002.

[48] J. Castro, "Minimum-distance controlled perturbation methods for large-scale tabular data protection," *European Journal of Operational Research*, vol. 171, 2004.

[49] K. Chaudhuri and N. Mishra, "When random sampling preserves privacy," in *Proceedings of the International Cryptology Conference*, 2006.

[50] B.-C. Chen, L. Chen, D. Musicant, and R. Ramakrishnan, "Learning from agggregate views," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE)*, 2006.

[51] B.-C. Chen, K. LeFevre, and R. Ramakrishnan, "PrivacySkyline: Privacy with multidimensional adversarial knowledge," in *Proceedings of the 33rd International Conference on Very Large Databases (VLDB)*, 2007.

[52] C.-Y. Chow and M. Mokbel, "Enabling private continuous queries for revealed user locations," in *Advances in Spatial and Temporal Databases*, 2007.

[53] R. Christensen, *Log-Linear Models and Logistic Regression*. Springer-Verlag, 1997.

[54] C. A. W. Citteur and L. C. R. J. Willenborg, "Public use microdata files: Current practices at national statistical bureaus," *Journal of Official Statistics*, vol. 9, no. 4, pp. 783–794, 1993.

[55] A. Cooper, "A survey of query log privacy-enhancing techniques from a policy perspective," *ACM Transactions on the Web*, vol. 2, pp. 19–27, 2008.

[56] D. A. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 3rd ed., 2008.

[57] L. H. Cox, J. P. Kelly, and R. Patil, "Balancing quality and confidentiality for multivariate tabular data," *Domingo-Ferrer J, Torra V, editors, Privacy in Statistical Databases*, 2004.

[58] L. H. Cox, S.-K. McDonald, and D. Nelson, "Confidentiality issues at the United States bureau of the census," *Journal of Official Statistics*, vol. 2, no. 2, pp. 135–160, 1986.

[59] L. H. Cox and L. V. Zayatz, "An Agenda for research in statistical disclosure limitation," *Journal of Official Statistics*, vol. 11, no. 2, pp. 205–220, 1995.

[60] T. Dalenius and S. Reiss, "Data swapping: A technique for disclosure control," *Journal of Statistical Planning and Inference*, vol. 6, pp. 73–85, 1982.

[61] N. N. Dalvi, G. Miklau, and D. Suciu, "Asymptotic conditional probabilities for conjunctive queries," in *Proceedings of the 10th International Conference on Database Theory (ICDT)*, 2005.

[62] N. N. Dalvi and D. Suciu, "Efficient query evaluation on probabilistic databases," *VLDB Journal*, vol. 16, no. 4, pp. 523–544, 2007.

[63] R. A. Dandekar, "Cost effective implementation of synthetic tabulation (a.k.a. controlled tabular adjustments) in legacy and new statistical data publication systems," *Joint ECE/Eurostat Work Session on Statistical Data Confidentiality, Luxembourg*, 2003.

[64] R. A. Dandekar, "Maximum utility-minimum information loss table server design for statistical disclosure control of tabular data," *Domingo-Ferrer J, Torra V, editors, Privacy in statistical databases*, 2004.

[65] R. A. Dandekar, M. Cohen, and N. Kirkendall, "Sensitive micro data protection using latin hypercube sampling technique," in *Inference Control in Statistical Databases, From Theory to Practice*, 2002.

[66] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th International Conference on World Wide Web (WWW)*, 2003.

[67] N. de Freitas and H. Kück, "Learning about individuals from group statistics," in *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI)*, pp. 332–339, 2005.

[68] A. de Waal and L. C. R. J. Willenborg, "Statistical disclosure control and sampling weights," *Journal of Official Statistics*, vol. 13, no. 4, pp. 417–434, 1997.

[69] D. Defays and M. N. Anwar, "Masking microdata using micro-aggregation," *Journal of Ofcial Statistics*, vol. 14, no. 4, 1998.

[70] D. E. Denning, P. J. Denning, and M. D. Schwartz, "The tracker: A threat to statistical database security," *ACM Transactions on Database Systems*, vol. 4, no. 1, pp. 76–96, 1979.

[71] D. E. Denning and J. Schlörer, "A fast procedure for finding a tracker in a statistical database," *ACM Transactions on Database Systems*, vol. 5, no. 1, pp. 88–102, 1980.

[72] P. Diaconis and B. Sturmfels, "Algebraic algorithms for sampling from conditional distributions," *Annals of Statistics*, vol. 1, pp. 363–397, 1998.

[73] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1–2, pp. 31–71, 1997.

[74] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, 2003.

[75] A. Dobra, "statistical tools for disclosure limitation in multiway contingency tables," PhD thesis, Carnegie Mellon University, 2002.

[76] A. Dobra, "Markov bases for decomposable graphical models," *Bernoulli*, vol. 9, no. 6, pp. 1093–1108, 2003.

[77] A. Dobra and S. E. Fienberg, *Assessing the Risk of Disclosure of Confidential Categorical Data*. Bayesian Statistics 7, Oxford University Press, 2000.

[78] A. Dobra and S. E. Fienberg, "Bounding entries in multi-way contingency tables given a set of marginal totals," in *Foundations of Statistical Inference: Proceedings of the Shoresh Conference 2000*, Springer Verlag, 2003.

[79] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, no. 1, 2002.

[80] J. Domingo-Ferrer, F. Sebe, and J. Castella-Roca, "On the security of noise addition for privacy in statistical databases," in *Privacy in Statistical Databases*, 2004.

[81] J. Domingo-Ferrer and V. Torra, "A quantitative comparison of disclosure control methods for microdata," in *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, 2001.

[82] W. Du, Z. Teng, and Z. Zhu, "Privacy-maxent: Integrating background knowledge in privacy quantification," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2008.

[83] G. T. Duncan and D. Lambert, "The risk of disclosure for microdata," *Journal of Business and Economic Statistics*, vol. 7, no. 2, 1989.

[84] Q. Duong, K. LeFevre, and M. Wellman, "Strategic modeling of information sharing among data privacy attackers," in *Quantitative Risk Analysis for Security Applications Workshop*, 2009.

[85] C. Dwork, "Differential privacy," in *ICALP*, 2006.

[86] C. Dwork, K. Kenthapadi, F. McSherry, and I. Mironov, "Our data, ourselves: Privacy via distributed noise generation," in *EUROCRYPT*, 2006.

[87] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*, pp. 265–284, 2006.

[88] C. Dwork, F. McSherry, and K. Talwar, "The price of privacy and the limits of LP decoding," in *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing (STOC)*, 2007.

[89] J. Edmonds, "Maximum matching and a polyhedron with 0–1 vertices," *Journal of Research of the National Bureau of Standards, Section B, Mathematical Sciences*, vol. 69, 1965.

[90] M. Elliot and A. Dale, "Scenarios of attack: The data intruder's perspective on statistical disclosure risk," *Netherlands Official Statistics*, vol. 14, pp. 6–10, 1999.

[91] A. Evfimievski, R. Fagin, and D. P. Woodruff, "Epistemic privacy," in *PODS*, 2008.

[92] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy-preserving data mining," in *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, 2003.

[93] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[94] C. Fang and E.-C. Chang, "Information leakage in optimal anonymized and diversified data," in *Information Hiding*, 2008.

[95] Federal Committee on Statistical Methodology. Statistical Policy Working Paper 22: Report on Statistical Disclosure Limitation Methodology, December 2005.

[96] I. Fellegi and A. Sunter, "A theory for record linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.

[97] I. P. Fellegi, "On the question of statistical confidentiality," *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 7–18, 1972.

[98] S. E. Fienberg, "Confidentiality and disclosure limitation methodology: Challenges for national statistics and statistical research," Technical Report 668, Carnegie Mellon University, 1997.

[99] S. E. Fienberg, U. E. Makov, and A. P. Sanil, "A bayesian approach to data disclosure: Optimal intruder behavior for continuous data," *Journal of Official Statistics*, vol. 13, no. 1, pp. 75–89, 1997.

[100] S. E. Fienberg, U. E. Makov, and R. J. Steele, "Disclosure limitation using perturbation and related methods for categorical data," *Journal of Official Statistics*, vol. 14, no. 4, pp. 485–502, 1998.

[101] S. E. Fienberg and J. McIntyre, "Data swapping: Variations on a theme by Dalenius and Reiss," in *Privacy in Statistical Databases*, pp. 14–29, 2004.

[102] S. E. Fienberg and A. B. Slavkovic, "Preserving the confidentiality of categorical statistical data bases when releasing information for association rules," *Data Mining Knowledge Discovery*, vol. 11, no. 2, pp. 155–180, 2005.

[103] L. Franconi and S. Polettini, "Individual risk estimation in $\mu$-argus: A review," in *Privacy in Statistical Databases*, 2004.

[104] L. Franconi and J. Stander, "A model-based method for disclosure limitation of business microdata," *The Statistician*, vol. 51, no. 1, pp. 51–61, 2002.

[105] D. Frankowski, D. Cosley, S. Sen, L. Terveen, and J. Riedl, "You are what you say: Privacy risks of public mentions," in *In Proceedings of the 29th SIGIR*, 2006.

[106] W. A. Fuller, "Masking procedures for microdata disclosure limitation," *Journal of Official Statistics*, vol. 9, no. 2, pp. 383–406, 1993.

[107] B. C. M. Fung, Ke. Wang, and P. S. Yu, "Top-down specialization for information and privacy preservation," in *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, 2005.

[108] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith, "Composition attacks and auxiliary information in data privacy," in *The 14th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2008.

[109] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized approach," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, 2005.

[110] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, 2008.

[111] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," in

*Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2008.

[112] G. Ghinita, P. Kalnis, and S. Skiadopoulis, "PRIVE: Anonymous location-based queries in distributed mobile systems," in *Proceedings of the 16th International World Wide Web Conference*, 2007.

[113] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast data anonymization with low information loss," in *Proceedings of the 34th International Conference on Very Large Databases*, 2007.

[114] G. Ghinita, Y. Tao, and P. Kalnis, "On the anonymization of sparse high-dimensional data," *IEEE 24th International Conference on Data Engineering (ICDE)*, 2008.

[115] A. Gionis and T. Tassa, "$k$-anonymization with minimal loss of information," in *TKDE*, 2008.

[116] S. Gomatam and A. F. Karr, "Distortion measures for categorical data swapping," Technical Report, National Institute of Statistical Sciences, Technical Report Number 131, 2003.

[117] J. Gouweleeuw, P. Kooiman, L. C. R. J. Willenborg, and P.-P. de Wolf, "Post randomisation for statistical disclosure control: Theory and implementation," *Journal of Official Statistics*, vol. 14, no. 4, 1998.

[118] A. J. Grove, J. Y. Halpern, and D. Koller, "Random worlds and maximum entropy," in *Logic in Computer Science*, pp. 22–33, 1992.

[119] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the First International Conference on Mobile Systems, Applications and Services*, 2003.

[120] M. Gruteser and B. Hoh, "On the anonymity of periodic location samples," in *Proceedings of the Second International Conference on Security in Pervasive Computing*, 2005.

[121] J. Hajek, Z. Sidak, and P. K. Sen, *Theory of Rank Tests*. Academic Press, 2nd ed., 1999.

[122] S. Hawla, L. Zayatz, and S. Rowland, "American factfinder: Disclosure limitation for the advanced query system," *Journal of Official Statistics*, vol. 20, no. 1, pp. 115–124, 2004.

[123] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," in *Proceedings of the 34th International Conference on Very Large Data Bases (VLDB)*, 2008.

[124] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving privacy in GPS traces via uncertainty-aware path cloaking," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2007.

[125] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, Dietrick A. Stephan John V. Pearson, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *Plos Genetics*, vol. 4, no. 8, 2008.

[126] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proceedings of the 23th ACM SIGMOD Conference on Management of Data*, June 2004.

[127] A. Hundepool, "The ARGUS software in the CASC-project," in *Privacy in Statistical Databases*, 2004.

[128] A. Hundepool and L. Willenborg, "$\mu$- and $\tau$-ARGUS: Software for statistical disclosure control," in *Proceedings of the Third International Seminar on Statistical Confidentiality*, 1996.

[129] J. T. Hwang, "Multiplicative errors-in-variables models with applications to recent data released by the U.S. department of energy," *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 680–688, 1986.

[130] S. Im, Z. W. Ras, and L.-S. Tsay, "Multi-granularity classification rule discovery using ERID," in *Proceedings of the 3rd International Conference on Rough Sets and Knowledge Technology (RSKT)*, pp. 491–499, 2008.

[131] T. Iwuchukwu and J. Naughton, "K-anonymization as spatial indexing: Toward scalable and incremental anonymization," in *Proceedings of the 33rd International Conference on Very Large Databases (VLDB)*, 2007.

[132] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[133] T. B. Jabine, "Statistical disclosure limitation practices at United States statistical agencies," *Journal of Official Statistics*, vol. 9, no. 2, pp. 427–454, 1993.

[134] W. Jin, K. LeFevre, and J. Patel, "An Online Framework for Publishing Dynamic Privacy-Sensitive GPS Traces," University of Michigan Technical Report CSE-TR-554-09, 2009.

[135] T. Johnsten and V. V. Raghavan, "Impact of decision-region based classification mining algorithms on database security," in *Proceedings of the IFIP WG 11.3 Thirteenth International Conference on Database Security*, pp. 177–191, Deventer, The Netherlands, The Netherlands, Kluwer, B.V., 2000.

[136] R. Jones, R. Kumar, B. Pang, and A. Tomkins, "I know what you did last summer — query logs and user privacy," in *Proceedings of the 16th Conference on Information and Knowledge Management (CIKM)*, 2007.

[137] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing Location-based identity inference on anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, 2007.

[138] H. Kargupta, S. Datta, Qi. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Proceedings of the International Conference on Data Mining*, 2003.

[139] A. F. Karr, C. N. Kohnen, A. Oganian, J. P. Reiter, and A. P. Sanil, "A framework for evaluating the utility of data altered to protect confidentiality," *The American Statistician*, vol. 60, pp. 224–232, 2006.

[140] A. B. Kennickell, "Multiple imputation and disclosure protection: The case of the 1995 survey of consumer finances," *Record Linkage Techniques*, pp. 248–267, 1997.

[141] D. Kifer, "Attacks on privacy and de Finetti's theorem," in *SIGMOD*, 2009.

[142] D. Kifer and J. Gehrke, "Injecting utility into anonymized datasets," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2006.

[143] J. Kim, "A method for limiting disclosure in microdata based on random noise and transformation," in *Proceedings of the Section on Survey Research Methods*, pp. 303–308, American Statistical Association, 1986.

[144] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas, "Releasing search queries and clicks privately," in *18th International World Wide Web Conference*, pp. 171–171, 2009.

[145] J. Krumm, "Inference attacks on location tracks," in *Proceedings of the 5th International Conference on Pervasive Computing*, 2007.

[146] R. Kumar, J. Novak, Bo. Pang, and A. Tomkins, "On anonymizing query logs via token-based hashing," in *Proceedings of the 16th International World Wide Web Conference (WWW)*, 2007.

[147] L. V. S. Lakshmanan, R. T. Ng, and G. Ramesh, "To do or not to do: The dilemma of disclosing anonymized data," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2005.

[148] D. Lambert, "Measures of disclosure risk and harm," *Journal of Official Statistics*, vol. 9, no. 2, pp. 313–331, 1993.

[149] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain $k$-anonymity," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2005.

[150] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional $k$-Anonymity," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE)*, 2006.

[151] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Workload-Aware Anonymization," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[152] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Workload-aware anonymization techniques for large-scale data sets," *ACM Transactions on Database Systems*, vol. 33, no. 3, 2008.

[153] R. Lenz, "Measuring the disclosure protection of micro aggregated business microdata. An analysis taking as an example the german structure of costs survey," *Journal of Official Statistics*, vol. 22, no. 4, pp. 681–710, 2006.

[154] F. Li, J. Sun, S. Papadimitriou, G. A. Mihaila, and I. Stanoi, "Hiding in the crowd: Privacy preservation on evolving streams through correlation tracking," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pp. 686–695, 2007.

[155] J. Li, Y. Tao, and X. Xiao, "Preservation of proximity privacy in publishing numeric sensitive data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2008.

[156] K. H. Li, T. E. Raghunathan, and D. B. Rubin, "Large-sample significance levels from multiply imputed data using moment-based statistics and an f reference distribution," *Journal of the American Statistical Association*, vol. 86, no. 416, pp. 1065–1073, 1991.

[157] N. Li, T. Li, and S. Venkatasubramanian, "$t$-Closeness: Privacy beyond $k$-anonymity and $l$-diversity," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2007.

[158] T. Li and N. Li, "Injector: Mining background knowledge for data anonymization," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2008.

[159] C. K. Liew, U. J. Choi, and C. J. Liew, "A data distortion by probability distribution," *ACM Transactions on Database Systems*, vol. 10, no. 3, pp. 395–411, 1985.

[160] R. J. A. Little, "Statistical analysis of masked data," *Journal of Official Statistics*, vol. 9, no. 2, pp. 407–426, 1993.

[161] F. Liu and R. J. A. Little, "Selective multiple imputation of keys for statistical disclosure control in microdata," in *ASA Proceedings of the Joint Statistical Meetings*, 2002.

[162] K. Liu, C. Giannella, and H. Kargupta, *A Survey of Attack Techniques on Privacy-preserving Data Perturbation Methods*. chapter 15, pp. 357–380, Springer, 2008.

[163] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2008.

[164] A. Machanavajjhala, "Defining and Enforcing Privacy in Data Sharing," PhD thesis, Cornell University, 2008.

[165] A. Machanavajjhala and J. Gehrke, "On the efficiency of checking perfect privacy," in *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 2006.

[166] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "$\ell$-Diversity: Privacy beyond $k$-anonymity," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2006.

[167] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: From theory to practice on the map," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2008.

[168] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "$\ell$-diversity: Privacy beyond $k$-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, 2007.

[169] B. Malin, "Betrayed by my shadow: Learning data identity via trail matching," *Journal of Privacy Technology*, p. 20050609001, 2005.

[170] B. Malin, "An evaluation of the current state of genomic data privacy protection and a roadmap for the future," *Journal of the American Medical Informatics Association*, vol. 12, no. 1, 2005.

[171] B. Malin, "Re-identification of familial database records," in *Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium*, 2006.

[172] B. Malin and L. Sweeney, "How (Not) to protect genomic data privacy in a distributed network: Using train re-identification to evaluate and design anonymity protection systems," *Journal of Biomedical Informatics*, vol. 37, no. 3, pp. 179–192, 2004.

[173] D. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Halpern, "Worst case background knowledge for privacy preserving data publishing," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2007.

[174] J. M. Mateo-Sanz, A. Martínez-Ballesté, and J. Domingo-Ferrer, "Fast generation of accurate synthetic microdata," in *Privacy in Statistical Databases*, 2004.

[175] P. McCullagh and J. A. Nelder, *Generalized Linear Models*. Chapman & Hall/CRC, 2nd ed., 1989.

[176] X.-L. Meng and D. B. Rubin, "Performing likelihood ratio tests with multiply-imputed data sets," *Biometrika*, vol. 79, no. 1, pp. 103–111, 1992.

[177] M. M. Meyer and J. B. Kadane, "Evaluation of a reconstruction of the adjusted 1990 census for Florida," *Journal of Official Statistics*, vol. 13, no. 2, pp. 103–112, 1997.

[178] A. Meyerson and R. Williams, "On the complexity of optimal $k$-anonymity," in *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 2004.

[179] G. Miklau and D. Suciu, "A formal analysis of information disclosure in data exchange," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2004.

[180] M. Mokbel, C. Chow, and W. Aref, "The new casper: Query processing for location services without compromising privacy," in *Proceedings of the 32nd International Conference on Very Large Databases (VLDB)*, 2006.

[181] R. A. Moore, "Controlled data-swapping techniques for masking public use microdata sets," Technical Report, U.S. Bureau of the Census, 1996.

[182] K. Muralidhar and R. Sarathy, "A theoretical basis for perturbation methods," *Statistics and Computing*, vol. 13, no. 4, pp. 329–335, 2003.

[183] K. Muralidhar and R. Sarathy, "A comparison of multiple imputation and data perturbation for masking numerical variables," *Journal of Official Statistics*, vol. 22, no. 3, pp. 507–524, 2006.

[184] M. Murugesan and C. Clifton, "Providing privacy through plausibly deniable search," in *Proceedings of the 9th SIAM International Conference on Data Mining (SDM)*, pp. 768–779, 2009.

[185] D. R. Musicant, J. M. Christensen, and J. F. Olson, "Supervised learning by training on aggregate outputs," in *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*, pp. 252–261, 2007.

[186] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE Symposium on Security and Privacy*, 2008.

[187] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *IEEE Symposium on Security and Privacy*, 2009.

[188] M. E. Nergiz, M. Atzori, and C. W. Clifton, "Hiding the presence of individuals from shared databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2007.

[189] M. E. Nergiz, M. Atzori, and Y. Saygin, "Toward trajectory anonymization: A generalization-based approach," in *Proceedings of the 2nd SIGSPATIAL ACM GIS International Workshop on Security and Privacy in GIS and LBS*, 2008.

[190] M. E. Nergiz and C. Clifton, "Thoughts on $k$-anonymization," *Data & Knowledge Engineering*, vol. 63, no. 3, pp. 622–645, 2007.

[191] Netflix. The Netflix Prize Rules: http://www.netflixprize.com//rules.

[192] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *39th ACM Symposium on Theory of Computing (STOC)*, 2007.

[193] J. Novak, P. Raghavan, and A. Tomkins, "Anti-aliasing on the web," in *Proceedings of the 13th International Conference on World Wide Web (WWW)*, 2004.

[194] A. Ohrn and L. Ohno-Machado, "Using Boolean reasoning to anonymize databases," *Artificial Intelligence in Medicine*, vol. 15, no. 3, pp. 235–254, 1999.

[195] G. Paass, "Disclosure risk and disclosure avoidance for microdata," *Journal of Business & Economic Statistics*, vol. 6, no. 4, pp. 487–500, October 1988.

[196] M. A. Palley and J. S. Simonoff, "The use of regression methodology for the compromise of confidential information in statistical databases," *ACM Transactions on Database Systems*, vol. 12, no. 4, pp. 593–608, 1987.

[197] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing (EMNLP)*, 2002.

[198] S. Papadimitriou, F. Li, G. Kollios, and P. S. Yu, "Time series compressibility and privacy," in *VLDB*, 2007.

[199] J. B. Paris, *The Uncertain Reasoner's Companion*. Cambridge University Press, 1994.

[200] H. Park and K. Shim, "Approximation algorithms for $k$-anonymity," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2007.

[201] J. Pei, J. Xu, Z. Wang, W. Wang, and Ke. Wang, "Maintaining $k$-anonymity against incremental updates," in *SSDBM*, 2007.

[202] B. Poblete, M. Spiliopoulou, and R. Baeza-Yates, "Website privacy preservation for query log publishing," in *Proceedings of the 1st ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD (PinKDD)*, 2007.

[203] S. Polettini, "Maximum entropy simulation for microdata protection," *Statistics and Computing*, vol. 13, pp. 307–320, 2003.

[204] S. Polettini and J. Stander, "A comment on "A theoretical basis for perturbation methods" by Krishnamurty Muralidhar and Rathindra Sarathy," *Statistics and Computing*, vol. 13, no. 4, pp. 337–338, 2003.

[205] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le, "Estimating labels from label proportions," in *Proceedings of the 25th International Conference Machine Learning (ICML)*, pp. 776–783, 2008.

[206] T. E. Raghunathan, J. M. Lepkowski, J. V. Hoewyk, and P. Solenberger, "A multivariate technique for multiply imputing missing values using a sequence of regression models," *Survey Methodology*, vol. 27, no. 1, pp. 85–95, 2001.

[207] T. E. Raghunathan, J. P. Reiter, and D. B. Rubin, "Multiple imputation for statistical disclosure limitation," *Journal of Official Statistics*, vol. 19, pp. 1–16, 2003.

[208] G. Ramesh, "Can attackers learn from samples?," in *2nd VLDB Workshop on Secure Data Management (SDM)*, 2005.

[209] J. N. K. Rao, "On variance estimation with imputed survey data," *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 499–506, 1996.
[210] V. Rastogi, M. Hay, G. Miklau, and D. Suciu, "Relationship privacy: Output perturbation for queries with joins," in *PODS*, 2009.
[211] V. Rastogi, D. Suciu, and S. Hong, "The boundary between privacy and utility in data publishing," Technical Report, University of Washington, 2007.
[212] V. Rastogi, D. Suciu, and S. Hong, "The boundary between privacy and utility in data publishing," in *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB)*, 2007.
[213] J. P. Reiter, "Inference for partially synthetic, public use microdata sets," *Survey Methodology*, vol. 29, no. 2, pp. 181–188, 2003.
[214] J. P. Reiter, "New approaches to data dissemination: A glimpse into the future (?)," *Chance*, vol. 17, no. 3, pp. 12–16, 2004.
[215] J. P. Reiter, "Simultaneous use of multiple imputation for missing data and disclosure limitation," *Survey Methodology*, vol. 30, pp. 235–242, 2004.
[216] J. P. Reiter, "Estimating risks of identification disclosure in microdata," *Journal of the American Statistical Association*, vol. 100, pp. 1103–1113, 2005.
[217] J. P. Reiter, "Significance tests for multi-component estimands from multiply-imputed, synthetic microdata," *Journal of Statistical Planning and Inference*, vol. 131, pp. 365–377, 2005.
[218] J. P. Reiter, "Using cart to generate partially synthetic public use microdata," *Journal of Official Statistics*, vol. 21, no. 3, pp. 441–462, 2005.
[219] Research Data Centres of the Federal Statistical Office and the Statistical Offices of the Lander http://www.forschungsdatenzentrum.de/en/index.asp, June 9, 2009.
[220] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 2005.
[221] D. B. Rubin, "Discussion statistical disclosure limitation," *Journal of Official Statistics*, vol. 9, no. 2, 1993.
[222] D. B. Rubin, "Multiple imputation after 18+ years," *Journal of the American Statistical Association*, vol. 91, pp. 473–489, 1996.
[223] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*. Wiley-Interscience, 2004.
[224] D. B. Rubin and N. Schenker, "Multiple imputation for interval estimation from simple random samples with ignorable nonresponse," *Journal of the American Statistical Association*, vol. 81, pp. 366–374, 1986.
[225] S. Ruggles Secure Data Laboratories: The U.S. Census Bureau Model.
[226] P. Samarati, "Protecting respondents' identities in microdata release," in *Transactions on Knowledge and Data Engineering*, pp. 1010–1027, 2001.
[227] J. A. Sanchez, J. Urrutia, and E. Ripoll, "Trade-off between disclosure risk and information loss using multivariate microaggregation: A case study on business data," in *Privacy in Statistical Databases*, pp. 307–322, 2004.
[228] J. L. Schafer, "Multiple imputation: A primer," *Statistical Methods in Medical Research*, vol. 8, no. 1, pp. 3–15, 1999.
[229] M. J. Schervish, *Theory of Statistics*. Springer, 1995.
[230] J. Schlorer, "Identification and retrieval of personal records from a statistical bank," in *Methods of Information in Medicine*, 1975.

[231] G. A. F. Seber and A. J. Lee, *Linear Regression Analysis*. John Wiley and Son, 2003.
[232] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, 1949.
[233] P. K. Shivaswamy, C. Bhattacharyya, and A. J. Smola, "Second order cone programming approaches for handling missing and uncertain data," *Journal of Machine Learning Research*, vol. 7, pp. 1283–1314, 2006.
[234] C. Skinner, C. Marsh, S. Openshaw, and C. Wymer, "Disclosure control for census microdata," *Journal of Official Statistics*, vol. 10, no. 1, pp. 31–51, 1994.
[235] A. Slavkovic and S. E. Fienberg, "Bounds for cell entries in two-way tables given conditional relative frequencies," in *Privacy in Statistical Databases*, 2004.
[236] N. L. Spruill, "Measures of confidentiality," *Statistics of Income and Related Administrative Record Research*, pp. 131–136, 1982.
[237] K. Stoffel and T. Studer, "Provable data privacy," in *Proceedings of the International Conference on Database and Expert Systems Applications (DEXA)*, pp. 324–332, 2005.
[238] L. Sweeney, "Datafly: A system for providing anonymity in medical data," in *DBSec*, pp. 356–381, 1997.
[239] L. Sweeney, "Guaranteeing anonymity when sharing medical data, the datafly system," *Journal of the American Medical Informatics Association*, 1997.
[240] L. Sweeney, "Uniqueness of simple demographics in the U.S. population," Technical Report, Carnegie Mellon University, 2000.
[241] L. Sweeney, "$k$-anonymity: A model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.
[242] A. Takemura, "Local recoding and record swapping by maximum weight matching for disclosure control of microdata sets," *Journal of Official Statistics*, vol. 18, 2002.
[243] P. Tendick, "Optimal noise addition for preserving confidentiality in multivariate data," *Journal of Statistical Planning and Inference*, vol. 27, no. 3, pp. 341–353, March 1991.
[244] B. J. Tepping, "A model for optimum linkage of records," *Journal of the American Statistical Association*, vol. 63, no. 324, pp. 1321–1332, 1968.
[245] M. Terrovitis and M. Mamoulis, "Privacy preservation in the publication of trajectories," in *Proceedings of the 9th International Conference on Mobile Data Management (MDM)*, 2008.
[246] M. Terrovitis, N. Mamoulis, and P. Kalnis, "Privacy-preserving anonymization of set-valued data," in *Proceedings of the 34th International Conference on Very Large Databases (VLDB)*, 2008.
[247] The Avatar Project at IBM: http://www.almaden.ibm.com/cs/projects/avatar/.
[248] The Federal Death Penalty System A Statistical Survey (1988–2000), http://www.usdoj.gov/dag/pubdoc/dpsurvey.html, September 12, 2000.
[249] The MystiQ Project at University of Washington: http://www.cs.washington.edu/homes/suciu/project-mystiq.html.

[250] The ORION database system at Purdue University: http://orion.cs. purdue.edu/.
[251] The Trio at Stanford University: http://infolab.stanford.edu/trio/.
[252] J.-A. Ting, A. D'Souza, and S. Schaal, "Bayesian regression with input noise for high dimensional data," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 937–944, 2006.
[253] V. Torra, "Microaggregation for categorical variables: A median based approach," *Privacy in Statistical Databases*, 2004.
[254] A. Torres, "Contribucions a la Microagregacio per a la Proteccio de Dades Estadistiques (Contributions to the Microaggregation for the Statistical Data Protection," PhD thesis, Universitat Politecnica de Catalunya, 2003.
[255] M. Trottini and S. E. Fienberg, "Modelling user uncertainty for disclosure risk and data utility," *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 511–527, 2002.
[256] U.S. Census Bureau Center for Economic Studies http://www.ces.census.gov, June 9, 2009.
[257] U.S. Census Bureau Statement of Confidentiality http://factfinder.census.gov/jsp/saff/SAFFInfo.jsp?_pageId=su5_confidentiality, Retrieved June 9, 2009.
[258] S. van Buuren and K. Oudshoom, "Flexible multivariate imputation by mice," Technical Report, Netherlands Organization for Applied Scientific Research (TNO) TNO report PG/VGZ/99.054, 1999.
[259] K. Wang, B. Fung, and P. Yu, "Template-based privacy preservation in classification problems," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, November 2005.
[260] K. Wang, B. Fung, and P. Yu, "Anonymizing sequential releases," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.
[261] K. Wang, P. S. Yu, and S. Chakraborty, "Bottom-up generalization: A data mining solution to privacy protection," in *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM)*, 2004.
[262] S. L. Warner, "Randomized Response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, 1965.
[263] L. Willenborg and T. de Waal, *Statistical Disclosure Control in Practice*. Springer-Verlag, 1996.
[264] L. Willenborg and T. de Waal, *Elements of Statistical Disclosure Control*. Springer, 2000.
[265] W. E. Winkler, "The state of record linkage and current research problems," Technical Report, Statistical Research Division, U.S. Census Bureau, 1999.
[266] W. E. Winkler, "Single-ranking micro-aggregation and re-identification," Techncial Report, U.S. Bureau of the Census Statistical Research Division, 2002.
[267] W. E. Winkler, "Using simulated annealing for $k$-anonymity," *Research Report Series (Statistics #2002-7), U. S. Census Bureau*, 2002.
[268] W. E. Winkler, "Masking and re-identification methods for public-use microdata: Overview and research problems," Research Report #2004-06, U.S. Bureau of the Census, 2004.

[269] W. E. Winkler, "Re-identification methods for masked microdata," *Privacy in Statistical Databases*, 2004.

[270] W. E. Winkler, "Overview of record linkage and current research directions," Technical Report, U.S. Bureau of the Census, 2005.

[271] R. Wong, A. Fu, K. Wang, and J. Pei, "Minimality attack in privacy preserving data publishing," in *Proceedings of the 33rd International Conference on Very Large Databases (VLDB)*, 2007.

[272] Workshop on Data Confidentiality http://dcws.stat.cmu.edu/zayatz.htm, September 6–7, 2007.

[273] X. Xiao and Y. Tao, "Anatomy: Simple and effective privacy preservation," in *Proceedings of the 32nd International Conference on Very Large Databases (VLDB)*, 2006.

[274] X. Xiao and Y. Tao, "Personalized Privacy Preservation," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2006.

[275] X. Xiao and Y. Tao, "$m$-invariance: Towards privacy preserving re-publication of dynamic datasets," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2007.

[276] H. Xiong, M. Steinbach, and V. Kumar, "Privacy leakage in multi-relational databases: A semi-supervised learning perspective," *VLDB Journal*, vol. 15, no. 4, pp. 388–402, 2006.

[277] S. Xu and X. Ye, "Risk & distortion based $k$-anonymity," *Information Security Applications*, 2008.

[278] C. Yao, X. S. Wang, and S. Jajodia, "Checking for $k$-anonymity violation by views," in *Proceedings of the 31st International Conference on Very Large Databases (VLDB)*, pp. 910–921, 2005.

[279] J. Zhang and V. Honavar, "Learning decision tree classifiers from attribute value taxonomies and partially specified data," in *Proceedings of International Conference Machine Learning (ICML)*, 2003.

[280] J. Zhang, D.-K. Kang, A. Silvescu, and V. Honavar, "Learning accurate and concise naïve Bayes classifiers from attribute value taxonomies and data," *Knowledge Information Systems*, vol. 9, no. 2, pp. 157–179, 2006.

[281] L. Zhang, S. Jajodia, and A. Brodsky, "Information disclosure under realistic assumptions: Privacy versus optimality," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, 2007.

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

[283] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Proceedings of Privacy, Security and Trust in KDD Workshop*, 2007.

[284] E. Zheleva and L. Getoor, "To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles," in *Proceedings of the World Wide Web Conference*, 2009.

[285] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhoos attacks," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2008.