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

k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY[1]

LATANYA SWEENEY

School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
E-mail: latanya@cs.cmu.edu

Received May 2002

Consider a data holder, such as a hospital or a bank, that has a privately held collection of person-specific, field structured data. Suppose the data holder wants to share a version of the data with researchers. How can a data holder release a version of its private data with scientific guarantees that the individuals who are the subjects of the data cannot be re-identified while the data remain practically useful? The solution provided in this paper includes a formal protection model named *k*-anonymity and a set of accompanying policies for deployment. A release provides *k*-anonymity protection if the information for each person contained in the release cannot be distinguished from at least *k*-1 individuals whose information also appears in the release. This paper also examines re-identification attacks that can be realized on releases that adhere to *k*-anonymity unless accompanying policies are respected. The *k*-anonymity protection model is important because it forms the basis on which the real-world systems known as Datafly, µ-Argus and *k*-Similar provide guarantees of privacy protection.

Keywords: data anonymity, data privacy, re-identification, data fusion, privacy.

1. Introduction

Society is experiencing exponential growth in the number and variety of data collections containing person-specific information as computer technology, network connectivity and disk storage space become increasingly affordable. Data holders, operating autonomously and with limited knowledge, are left with the difficulty of releasing information that does not compromise privacy, confidentiality or national interests. In many cases the survival of the database itself depends on the data holder's ability to produce anonymous data because not releasing such information at all may diminish the need for the data, while on the other hand, failing to provide proper protection within a release may create circumstances that harm the public or others.

[1] This paper significantly amends and substantially expands the earlier paper "Protecting privacy when disclosing information: *k*-anonymity and its enforcement through generalization and suppression" (with Samarati) submitted to IEEE Security and Privacy 1998, and extends parts of my Ph.D. thesis "Computational Disclosure Control: A primer on data privacy protection" at the Massachusetts Institute of Technology 2001.

So a common practice is for organizations to release and receive person-specific data with all explicit identifiers, such as name, address and telephone number, removed on the assumption that anonymity is maintained because the resulting data look anonymous. However, in most of these cases, the remaining data can be used to re-identify individuals by linking or matching the data to other data or by looking at unique characteristics found in the released data.

In an earlier work, experiments using 1990 U.S. Census summary data were conducted to determine how many individuals within geographically situated populations had combinations of demographic values that occurred infrequently [1]. Combinations of few characteristics often combine in populations to uniquely or nearly uniquely identify some individuals. For example, a finding in that study was that 87% (216 million of 248 million) of the population in the United States had reported characteristics that likely made them unique based only on {*5-digit ZIP*[2], *gender*, *date of birth*}. Clearly, data released containing such information about these individuals should not be considered anonymous. Yet, health and other person-specific data are often publicly available in this form. Below is a demonstration of how such data can be re-identified.

Example 1. Re-identification by linking

The National Association of Health Data Organizations (NAHDO) reported that 37 states in the USA have legislative mandates to collect hospital level data and that 17 states have started collecting ambulatory care data from hospitals, physicians offices, clinics, and so forth [2]. The leftmost circle in Figure 1 contains a subset of the fields of information, or *attributes*, that NAHDO recommends these states collect; these attributes include the patient's ZIP code, birth date, gender, and ethnicity.

In Massachusetts, the Group Insurance Commission (GIC) is responsible for purchasing health insurance for state employees. GIC collected patient-specific data with nearly one hundred attributes per encounter along the lines of the those shown in the leftmost circle of Figure 1 for approximately 135,000 state employees and their families. Because the data were believed to be anonymous, GIC gave a copy of the data to researchers and sold a copy to industry [3].

For twenty dollars I purchased the voter registration list for Cambridge Massachusetts and received the information on two diskettes [4]. The rightmost circle in Figure 1 shows that these data included the name, address, ZIP code, birth date, and gender of each voter. This information can be linked using ZIP code, birth date and gender to the medical information, thereby

[2] In the United States, a ZIP code refers to the postal code assigned by the U.S. Postal Service. Typically 5-digit ZIP codes are used, though 9-digit ZIP codes have been assigned. A 5-digit code is the first 5 digits of the 9-digit code.

linking diagnosis, procedures, and medications to particularly named individuals.

For example, William Weld was governor of Massachusetts at that time and his medical records were in the GIC data. Governor Weld lived in Cambridge Massachusetts. According to the Cambridge Voter list, six people had his particular birth date; only three of them were men; and, he was the only one in his 5-digit ZIP code.

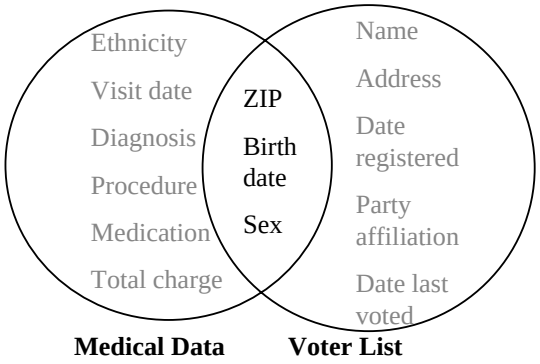

Figure 1 Linking to re-identify data

The example above provides a demonstration of re-identification by directly linking (or "matching") on shared attributes. The work presented in this paper shows that altering the released information to map to many possible people, thereby making the linking ambiguous, can thwart this kind of attack. The greater the number of candidates provided, the more ambiguous the linking, and therefore, the more anonymous the data.

2. Background

The problem of releasing a version of privately held data so that the individuals who are the subjects of the data cannot be identified is not a new problem. There are existing works in the statistics community on statistical databases and in the computer security community on multi-level databases to consider. However, none of these works provide solutions to the broader problems experienced in today's data rich setting.

2.1. Statistical databases

Federal and state statistics offices around the world have traditionally been entrusted with the release of statistical information about all aspects of the populace [5]. But like other data holders, statistics offices are also facing tremendous demand for person-specific data for applications such as data mining,

cost analysis, fraud detection and retrospective research. But many of the established statistical database techniques, which involve various ways of adding noise [6] to the data while still maintaining some statistical invariant [7, 8], often destroy the integrity of records, or *tuples*, and so, for many new uses of data, these established techniques are not appropriate. Willenborg and De Waal [9] provide more extensive coverage of traditional statistical techniques.

2.2. Multi-level databases

Another related area is aggregation and inference in multi-level databases [10, 11, 12, 13, 14, 15] which concerns restricting the release of lower classified information such that higher classified information cannot be derived. Denning and Lunt [16] described a multilevel relational database system (MDB) as having data stored at different security classifications and users having different security clearances.

Su and Ozsoyoglu formally investigated inference in MDB. They showed that eliminating precise inference compromise due to functional dependencies and multi-valued dependencies is NP-complete. By extension to this work, the precise elimination of all inferences with respect to the identities of the individuals whose information is included in person-specific data is typically impossible to guarantee. Intuitively this makes sense because the data holder cannot consider a priori every possible attack. In trying to produce anonymous data, the work that is the subject of this paper seeks to primarily protect against known attacks. The biggest problems result from inferences that can be drawn after linking the released data to other knowledge, so in this work, it is the ability to link the result to foreseeable data sources that must be controlled.

Many aggregation inference problems can be solved by database design, but this solution is not practical in today's data rich setting. In today's environment, information is often divided and partially replicated among multiple data holders and the data holders usually operate autonomously in making decisions about how data will be released. Such decisions are typically made locally with incomplete knowledge of how sensitive other holders of the information might consider replicated data. For example, when somewhat aged information on joint projects is declassified differently by the Department of Defense than by the Department of Energy, the overall declassification effort suffers; using the two partial releases, the original may be reconstructed in its entirety. In general, systems that attempt to produce anonymous data must operate without the degree of omniscience and level of control typically available in the traditional aggregation problem.

In both aggregation and MDB, the primary technique used to control the flow of sensitive information is *suppression*, where sensitive information and all information that allows the inference of sensitive information are simply not released. Suppression can drastically reduce the quality of the data, and in the

case of statistical use, overall statistics can be altered, rendering the data practically useless. When protecting national interests, not releasing the information at all may be possible, but the greatest demand for person-specific data is in situations where the data holder must provide adequate protections while keeping the data useful, such as sharing person-specific medical data for research purposes.

2.3. Computer security is not privacy protection

An area that might appear to have a common ancestry with the subject of this paper is access control and authentication, which are traditional areas associated with computer security. Work in this area ensures that the recipient of information has the authority to receive that information. While access control and authentication protections can safeguard against direct disclosures, they do not address disclosures based on inferences that can be drawn from released data. The more insidious problem in the work that is the subject of this paper is not so much whether the recipient can get access or not to the information as much as what values will constitute the information the recipient will receive. A general doctrine of the work presented herein is to release all the information but to do so such that the identities of the people who are the subjects of the data (or other sensitive properties found in the data) are protected. Therefore, the goal of the work presented in this paper lies outside of traditional work on access control and authentication.

2.4. Multiple queries can leak inference

Rückschluss, Folgerung, Deduktion

Denning [17] and others [18, 19] were among the first to explore inferences realized from multiple queries to a database. For example, consider a table containing only (physician, patient, medication). A query listing the patients seen by each physician, i.e., a relation R(physician, patient), may not be sensitive. Likewise, a query itemizing medications prescribed by each physician may also not be sensitive. But the query associating patients with their prescribed medications may be sensitive because medications typically correlate with diseases. One common solution, called query restriction, prohibits queries that can reveal sensitive information. This is effectively realized by suppressing all inferences to sensitive data. In contrast, this work poses a real-time solution to this problem by advocating that the data be first rendered sufficiently anonymous, and then the resulting data used as the basis on which queries are processed. Doing so typically retains far more usefulness in the data because the resulting release is often less distorted.

In summary, the dramatic increase in the availability of person-specific data from autonomous data holders has expanded the scope and nature of inference control problems and exasperated established operating practices. The goal of this work

is to provide a model for understanding, evaluating and constructing computational systems that control inferences in this setting.

3. Methods

The goal of this section is to provide a formal framework for constructing and evaluating algorithms and systems that release information such that the released information limits what can be revealed about properties of the entities that are to be protected. For convenience, I focus on person-specific data, so the entities are people, and the property to be protected is the identity of the subjects whose information is contained in the data. However, other properties could also be protected. The formal methods provided in this paper include the *k*-anonymity protection model. The real-world systems Datafly [20], μ-Argus [21] and *k*-Similar [22] motivate this approach.

Unless otherwise stated, the term *data* refers to person-specific information that is conceptually organized as a table of rows (or records) and columns (or fields). Each row is termed a *tuple*. A tuple contains a relationship among the set of values associated with a person. Tuples within a table are not necessarily unique. Each column is called an *attribute* and denotes a field or semantic category of information that is a set of possible values; therefore, an attribute is also a domain. Attributes within a table are unique. So by observing a table, each row is an ordered *n*-tuple of values $<d_1, d_2, \ldots, d_n>$ such that each value d_j is in the domain of the *j*-th column, for $j=1, 2, \ldots, n$ where *n* is the number of columns. In mathematical set theory, a relation corresponds with this tabular presentation, the only difference is the absence of column names. Ullman provides a detailed discussion of relational database concepts [23].

Definition 1. Attributes

Let $B(A_1,\ldots,A_n)$ be a *table* with a finite number of tuples. The finite set of *attributes* of B are $\{A_1,\ldots,A_n\}$.

Given a table $B(A_1,\ldots,A_n)$, $\{A_i,\ldots,A_j\} \subseteq \{A_1,\ldots,A_n\}$, and a tuple $t \in B$, I use $t[A_i,\ldots,A_j]$ to denote the sequence of the values, $v_i,\ldots,v_j$, of $A_i,\ldots,A_j$ in *t*. I use $B[A_i,\ldots,A_j]$ to denote the projection, maintaining duplicate tuples, of attributes $A_i,\ldots A_j$ in B.

Throughout the remainder of this work each tuple is assumed to be specific to one person and no two tuples pertain to the same person. This assumption simplifies discussion without loss of applicability.

Offenlegung, Aufdeckung

To draw an *inference* is to come to believe a new fact on the basis of other information. ~~A~~ *disclosure* means that explicit or inferable information about a person was released that was not intended. This definition may not be consistent with colloquial use but is used in this work consistent with its meaning in

statistical disclosure control. So, *disclosure control* attempts to identify and limit disclosures in released data. Typically the goal of disclosure control with respect to person-specific data is to ensure that released data are sufficiently anonymous.

Let me be more specific about how properties are selected and controlled. Recall the linking example shown in Figure 1. In that case, the need for protection centered on limiting the ability to link released information to other external collections. So the properties to be controlled are operationally realized as attributes in the privately held collection. The data holder is expected to identify all attributes in the private information that could be used for linking with external information. Such attributes not only include explicit identifiers such as name, address, and phone number, but also include attributes that in combination can uniquely identify individuals such as birth date and gender. The set of such attributes has been termed a *quasi-identifier* by Dalenius [24]. So operationally, a goal of this work is to release person-specific data such that the ability to link to other information using the quasi-identifier is limited.

Definition 2. Quasi-identifier

Given a population of entities U, an entity-specific table $T(A_1,\ldots,A_n)$, f_c: $U \rightarrow T$ and f_g: $T \rightarrow U'$, where $U \subseteq U'$. A quasi-identifier of T, written Q_T, is a set of attributes $\{A_i,\ldots,A_j\} \subseteq \{A_1,\ldots,A_n\}$ where: $\exists p_i \in U$ such that $f_g(f_c(p_i)[Q_T]) = p_i$.

Example 2. Quasi-identifier

Let V be the voter-specific table described earlier in Figure 1 as the voter list. A quasi-identifier for V, written Q_V, is {*name*, *address*, *ZIP*, *birth date*, *gender*}.

Linking the voter list to the medical data as shown in Figure 1, clearly demonstrates that {*birth date*, *ZIP*, *gender*} $\subseteq Q_V$. However, {*name*, *address*} $\subseteq Q_V$ because these attributes can also appear in external information and be used for linking.

In the case of anonymity, it is usually publicly available data on which linking is to be prohibited and so attributes which appear in private data and also appear in public data are candidates for linking; therefore, these attributes constitute the quasi-identifier and the disclosure of these attributes must be controlled. It is believed that these attributes can be easily identified by the data holder.

Assumption (quasi-identifier).
The data holder can identify attributes in his private data that may also appear in external information and therefore, can accurately identify quasi-identifiers.

Consider an instance where this assumption is incorrect; that is, the data holder misjudges which attributes are sensitive for linking. In this case, the released data may be less anonymous than what was required, and as a result, individuals may be more easily identified. Clearly, this risk cannot be perfectly resolved by the data holder because the data holder cannot always know what each recipient of the data knows but policies and contracts, which lie outside the algorithms, can help. Also, the data holder may find it necessary to release data that are only partially anonymous. Again, policies, laws and contracts can provide complementary protections. In the remainder of this work, I assume a proper quasi-identifier has been recognized.

As an aside, there are many ways to expand the notion of a quasi-identifier to provide more flexibility and granularity. Both the Datafly and μ-Argus systems weight the attributes of the quasi-identifier. For simplicity in this work, however, I consider a single quasi-identifier based on attributes, without weights, appearing together in an external table or in a possible join of external tables.

3.1. The *k*-anonymity protection model

In an earlier work, I introduced basic protection models termed *null-map*, *k-map* and *wrong-map* which provide protection by ensuring that released information map to no, *k* or incorrect entities, respectively [25]. To determine how many individuals each released tuple actually matches requires combining the released data with externally available data and analyzing other possible attacks. Making such a determination directly can be an extremely difficult task for the data holder who releases information. Although I can assume the data holder knows which data in PT also appear externally, and therefore what constitutes a quasi-identifier, the specific values contained in external data cannot be assumed. I therefore seek to protect the information in this work by satisfying a slightly different constraint on released data, termed the *k-anonymity* requirement. This is a special case of *k*-map protection where *k* is enforced on the released data.

Definition 3. *k*-anonymity

Let RT(A_1,...,A_n) be a table and QI_{RT} be the quasi-identifier associated with it. RT is said to satisfy *k*-anonymity if and only if each sequence of values in RT[QI_{RT}] appears with at least *k* occurrences in RT[QI_{RT}].

	Race	Birth	Gender	ZIP	Problem
t1	Black	1965	m	0214*	short breath
t2	Black	1965	m	0214*	chest pain
t3	Black	1965	f	0213*	hypertension
t4	Black	1965	f	0213*	hypertension
t5	Black	1964	f	0213*	obesity
t6	Black	1964	f	0213*	chest pain
t7	White	1964	m	0213*	chest pain
t8	White	1964	m	0213*	obesity
t9	White	1964	m	0213*	short breath
t10	White	1967	m	0213*	chest pain
t11	White	1967	m	0213*	chest pain

Figure 2 Example of *k*-anonymity, where *k*=2 and QI={*Race, Birth, Gender, ZIP*}

Example 3. Table adhering to *k*-anonymity

Figure 2 provides an example of a table T that adheres to *k*-anonymity. The quasi-identifier for the table is QI_T= {*Race*, *Birth*, *Gender*, *ZIP*} and *k*=2. Therefore, for each of the tuples contained in the table T, the values of the tuple that comprise the quasi-identifier appear at least twice in T. That is, for each sequence of values in $T[QI_T]$ there are at least *2* occurrences of those values in $T[QI_T]$. In particular, $t1[QI_T] = t2[QI_T]$, $t3[QI_T] = t4[QI_T]$, $t5[QI_T] = t6[QI_T]$, $t7[QI_T] = t8[QI_T] = t9[QI_T]$, and $t10[QI_T] = t11[QI_T]$.

Lemma.

Let $RT(A_1,...,A_n)$ be a table, $QI_{RT} = (A_i,\ldots, A_j)$ be the quasi-identifier associated with RT, $A_i,\ldots,A_j \subseteq A_1,\ldots,A_n$, and RT satisfy *k*-anonymity. Then, each sequence of values in $RT[A_x]$ appears with at least *k* occurrences in $RT[QI_{RT}]$ for $x=i,\ldots,j$.

Example 4. *k* occurrences of each value under *k*-anonymity

Table T in Figure 2 adheres to *k*-anonymity, where QI_T= {*Race*, *Birth*, *Gender*, *ZIP*} and *k*=2. Therefore, each value that appears in a value associated with an attribute of QI in T appears at least *k* times. |T[*Race* ="black"]| = 6. |T[*Race* ="white"]| = 5. |T[*Birth* ="1964"]| = 5. |T[*Birth* ="1965"]| = 4. |T[*Birth* ="1967"]| = 2. |T[*Gender* ="m"]| = 6. |T[*Gender* ="f"]| = 5. |T[*ZIP* ="0213*"]| = 9. And, |T[*ZIP* ="0214*"]| = 2.

It can be trivially proven that if the released data RT satisfies *k*-anonymity with respect to the quasi-identifier QI_{PT}, then the combination of the released data RT and the external sources on which QI_{PT} was based, cannot link on QI_{PT} or a subset of its attributes to match fewer than *k* individuals. This property holds provided that all attributes in the released table RT which are externally available in

combination (i.e., appearing together in an external table or in a possible join of external tables) are defined in the quasi-identifier QI_{PT} associated with the private table PT. This property does not guarantee individuals cannot be identified in RT; there may exist other inference attacks that could reveal the identities of the individuals contained in the data. However, the property does protect RT against inference from linking (by direct matching) to known external sources; and in this context, the solution can provide an effective guard against re-identifying individuals.

Race	ZIP
Asian	02138
Asian	02139
Asian	02141
Asian	02142
Black	02138
Black	02139
Black	02141
Black	02142
White	02138
White	02139
White	02141
White	02142

PT

Race	ZIP
Person	02138
Person	02139
Person	02141
Person	02142
Person	02138
Person	02139
Person	02141
Person	02142
Person	02138
Person	02139
Person	02141
Person	02142

GT1

Race	ZIP
Asian	02130
Asian	02130
Asian	02140
Asian	02140
Black	02130
Black	02130
Black	02140
Black	02140
White	02130
White	02130
White	02140
White	02140

GT2

Figure 3 Examples of *k*-anonymity tables based on PT

4. Attacks against k-anonymity

Even when sufficient care is taken to identify the quasi-identifier, a solution that adheres to *k*-anonymity can still be vulnerable to attacks. Three are described below. Fortunately, the attacks presented can be thwarted by due diligence to some accompanying practices, which are also described below.

4.1. Unsorted matching attack against *k*-anonymity

This attack is based on the order in which tuples appear in the released table. While I have maintained the use of a relational model in this discussion, and so the order of tuples cannot be assumed, in real-world use this is often a problem. It can be corrected of course, by randomly sorting the tuples of the solution table. Otherwise, the release of a related table can leak sensitive information.

Example 5. Unsorted matching attack

Tables GT1 and GT2 in Figure 3 are based on PT and adhere to *k*-anonymity, where QI_{PT} = {*Race*, *ZIP*} and *k*=2. The positions of the tuples in each table correspond to those in PT. If GT1 is released and a subsequent release of GT2 is then performed, then direct matching of tuples across the tables based

on tuple position within the tables reveals sensitive information. On the other hand, if the positions of the tuples within each table are randomly determined, both tables can be released.

4.2. Complementary release attack against *k*-anonymity

In the previous example, all the attributes were in the quasi-identifier. That is typically not the case. It is more common that the attributes that constitute the quasi-identifier are themselves a subset of the attributes released. As a result, when a table T, which adheres to *k*-anonymity, is released, it should be considered as joining other external information. Therefore, subsequent releases of the same privately held information must consider all of the released attributes of T a quasi-identifier to prohibit linking on T, unless of course, subsequent releases are based on T.

Example 6. Complementary release attack

Consider the private table PT in Figure 4. The tables GT1 and GT3 in Figure 5 are based on PT and adhere to *k*-anonymity, where *k*=2 and the quasi-identifier QI_{PT}={*Race*, *BirthDate*, *Gender*, *ZIP*}. Suppose table GT1 is released. If subsequently GT3 is also released, then the *k*-anonymity protection will no longer hold, even though the tuple positions are randomly determined in both tables. Linking GT1 and GT3 on {*Problem*} reveals the table LT shown in Figure 4. Notice how **[**white, 1964, male, 02138**]** and **[**white, 1965, female, 02139**]** are unique in LT and so, LT does not satisfy the *k*-anonymity requirement enforced by GT1 and GT3. This problem would not exist if GT3 used the quasi=identifier QI ∪ {*Problem*} or if GT1 had been the basis of GT3. In this latter case, no value more specific than it appears in GT1 would be subsequently released.

Race	BirthDate	Gender	ZIP	Problem
black	9/20/1965	male	02141	short of breath
black	2/14/1965	male	02141	chest pain
black	10/23/1965	female	02138	painful eye
black	8/24/1965	female	02138	wheezing
black	11/7/1964	female	02138	obesity
black	12/1/1964	female	02138	chest pain
white	10/23/1964	male	02138	short of breath
white	3/15/1965	female	02139	hypertension
white	8/13/1964	male	02139	obesity
white	5/5/1964	male	02139	fever
white	2/13/1967	male	02138	vomiting
white	3/21/1967	male	02138	back pain

PT

Race	BirthDate	Gender	ZIP	Problem
black	1965	male	02141	short of breath
black	1965	male	02141	chest pain
black	1965	female	02138	painful eye
black	1965	female	02138	wheezing
black	1964	female	02138	obesity
black	1964	female	02138	chest pain
white	1964	male	02138	short of breath
white	1965	female	02139	hypertension
white	1964	male	02139	obesity
white	1964	male	02139	fever
white	1967	male	02138	vomiting
white	1967	male	02138	back pain

LT

Figure 4 Private Table PT and linked table LT

Race	BirthDate	Gender	ZIP	Problem
black	1965	male	02141	short of breath
black	1965	male	02141	chest pain
person	1965	female	0213*	painful eye
person	1965	female	0213*	wheezing
black	1964	female	02138	obesity
black	1964	female	02138	chest pain
white	1964	male	0213*	short of breath
person	1965	female	0213*	hypertension
white	1964	male	0213*	obesity
white	1964	male	0213*	fever
white	1967	male	02138	vomiting
white	1967	male	02138	back pain

GT1

Race	BirthDate	Gender	ZIP	Problem
black	1965	male	02141	short of breath
black	1965	male	02141	chest pain
black	1965	female	02138	painful eye
black	1965	female	02138	wheezing
black	1964	female	02138	obesity
black	1964	female	02138	chest pain
white	1960-69	male	02138	short of breath
white	1960-69	human	02139	hypertension
white	1960-69	human	02139	obesity
white	1960-69	human	02139	fever
white	1960-69	male	02138	vomiting
white	1960-69	male	02138	back pain

GT3

Figure 5 Two *k*-anonymity tables based on PT in Figure 4 where *k*=2

4.3. Temporal attack against *k*-anonymity

Data collections are dynamic. Tuples are added, changed, and removed constantly. As a result, releases of generalized data over time can be subject to a temporal inference attack. Let table T_0 be the original privately held table at time t=0. Assume a *k*-anonymity solution based on T_0, which I will call table RT_0, is released. At time *t*, assume additional tuples were added to the privately held table T_0, so it comes T_t. Let RT_t be a *k*-anonymity solution based on T_t that is released at time *t*. Because there is no requirement that RT_t respect RT_0, linking the tables RT_0 and RT_t may reveal sensitive information and thereby compromise *k*-anonymity protection. As was the case in the previous example, to combat this problem, RT_0 should be considered as joining other external information. Therefore, either all of the attributes of RT_0 would be considered a quasi-identifier for subsequent releases, or subsequent releases themselves would be based on RT_0.

Example 7. Temporal attack

At time t_0, assume the privately held information is PT in Figure 4. As stated earlier, GT1 and GT3 in Figure 5 are *k*-anonymity solutions based on PT over the quasi-identifier QI_{PT}={*Race*, *BirthDate*, *Gender*, *ZIP*} where *k*=2. Assume GT1 is released. At a later time t_1, PT becomes PT_{t1}, which is PT ∪ {**[**`black`, `9/7/65`, `male`, `02139`, `headache`**]**, **[**`black`, `11/4/65`, `male`, `02139`, `rash`**]**}. Assume a *k*-anonymity solution based on PT is provided, and that it is called GT_{t1}. Assume this table contains GT3 in Figure 5; specifically, GT_{t1} = GT3 ∪ {**[**`black`, `1965`, `male`, `02139`, `headache`**]**, **[**`black`, `1965`, `male`, `02139`, `rash`**]**}. As was shown in the previous example, GT1 and GT3 can be linked on {*Problem*} to reveal unique tuples over QI_{PT}. Likewise, GT1 and GT_{t1} can be linked to reveal the same unique tuples. One way to combat this problem is to base k-anonymity solutions on

GT1 ∪ (PT_{t1} – PT). In that case, a result could be GT1 ∪ {**[**black, 1965, male, 02139, headache**]**, **[**black, 1965, male, 02139, rash**]**}, which does not compromise the distorted values in GT1.

5. Concluding remark

In this paper, I have presented the *k*-anonymity protection model, explored related attacks and provided ways in which these attacks can be thwarted.

Acknowledgments

First, I thank Vicenc Torra and Josep Domingo for their encouragement to write this paper. I also credit Pierangela Samarati for naming *k*-anonymity and thank her for recommending I engage the material in a formal manner and starting me in that direction in 1998. Finally, I am extremely grateful to the corporate and government members of the Laboratory for International Data Privacy at Carnegie Mellon University for providing me support and the opportunity to work on real-world data anonymity problems.

References

1 L. Sweeney, *Uniqueness of Simple Demographics in the U.S. Population*, LIDAP-WP4. Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA: 2000. Forthcoming book entitled, *The Identifiability of Data*.

2 National Association of Health Data Organizations, *A Guide to State-Level Ambulatory Care Data Collection Activities* (Falls Church: National Association of Health Data Organizations, Oct. 1996).

3 Group Insurance Commission testimony before the Massachusetts Health Care Committee. See *Session of the Joint Committee on Health Care, Massachusetts State Legislature*, (March 19, 1997).

4 Cambridge Voters List Database. *City of Cambridge, Massachusetts*. Cambridge: February 1997.

5 I. Fellegi. On the question of statistical confidentiality. *Journal of the American Statistical Association*, 1972, pp. 7-18.

6 J. Kim. A method for limiting disclosure of microdata based on random noise and transformation *Proceedings of the Section on Survey Research Methods of the American Statistical Association*, 370-374. 1986.

7 M. Palley and J. Siminoff. Regression methodology based disclosure of a statistical database *Proceedings of the Section on Survey Research Methods of the American Statistical Association* 382-387. 1986.

8 G. Duncan and R. Pearson. Enhancing access to data while protecting confidentiality: prospects for the future. *Statistical Science*, May, as Invited Paper with Discussion. 1991.

9 L. Willenborg and T. De Waal. *Statistical Disclosure Control in Practice*. Springer-Verlag, 1996.

10 T. Su and G. Ozsoyoglu. Controlling FD and MVD inference in multilevel relational database systems. *IEEE Transactions on Knowledge and Data Engineering*, 3:474--485, 1991.

11 M. Morgenstern. Security and Inference in multilevel database and knowledge based systems. *Proc. of the ACM SIGMOD* Conference, pages 357--373, 1987.

12 T. Hinke. Inference aggregation detection in database management systems. In *Proc. of the IEEE Symposium on Research in Security and Privacy*, pages 96-107, Oakland, 1988.

13 T. Lunt. Aggregation and inference: Facts and fallacies. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 102--109, Oakland, CA, May 1989.

14 X. Qian, M. Stickel, P. Karp, T. Lunt, and T. Garvey. Detection and elimination of inference channels in multilevel relational database systems. In *Proc. of the IEEE Symposium on Research in Security and Privacy*, pages 196--205, 1993.

15 T. Garvey, T. Lunt and M. Stickel. Abductive and approximate reasoning models for characterizing inference channels. *IEEE Computer Security Foundations Workshop*, **4**, 1991.

16 D. Denning and T. Lunt. A multilevel relational data model. In *Proc. of the IEEE Symposium on Research in Security and Privacy*, pages 220-234, Oakland, 1987.

17 D. Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.

18 D. Denning, P. Denning, and M. Schwartz. The tracker: A threat to statistical database security. *ACM Trans. on Database Systems*, 4(1):76--96, March 1979.

19 G. Duncan and S. Mukherjee. Microdata disclosure limitation in statistical databases: query size and random sample query control. In *Proc. of the 1991 IEEE Symposium on Research in Security and Privacy*, May 20-22, Oakland, California. 1991.

20 L. Sweeney. Guaranteeing anonymity when sharing medical data, the Datafly system. *Proceedings, Journal of the American Medical Informatics Association*. Washington, DC: Hanley & Belfus, Inc., 1997.

21 A. Hundepool and L. Willenborg. μ- and τ-argus: software for statistical disclosure control. *Third International Seminar on Statistical Confidentiality.* Bled: 1996.

22 L. Sweeney. Towards the optimal suppression of details when disclosing medical data, the use of sub-combination analysis. *Proceedings, MEDINFO 98*. International Medical Informatics Association. Seoul, Korea. North-Holland, 1998.

23 J. Ullman. *Principles of Database and Knowledge Base Systems*. Computer Science Press, Rockville, MD. 1988.

24 T. Dalenius. Finding a needle in a haystack – or identifying anonymous census record. *Journal of Official Statistics*, 2(3):329-336, 1986.

25 L. Sweeney, *Computational Data Privacy Protection*, LIDAP-WP5. Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA: 2000. Forthcoming book entitled, *A Primer on Providing Privacy in Data*.