

Betrayed By My Shadow: Learning Data Identity via Trail Matching

Bradley Malin

*Data Privacy Laboratory, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890, USA*

MALIN@PRIVACY.CS.CMU.EDU

Abstract

The term “re-identification” refers to the correct relation of seemingly anonymous data to explicitly identifying information, such as the name or address, of people who are the subjects of the data. Historically, methods for re-identification have been based on data released from a single data holder. This paper extends the concept to trail re-identification in which a person is related to seemingly anonymous data left behind at multiple locations, thus the data’s trail. The main premise behind these methods is that some locations capture, in addition to seemingly anonymous data, an individual’s explicitly identifying information and, subsequently, provide separate data releases of the unidentified and identified data. A single location’s releases appear unrelated; however, when multiple locations make such releases of information, common patterns in the data trails of two types of data can be used to discover relationships between them. The algorithms presented herein differ in the amount of completeness and multiplicity assumed in the data. We report experiments and successful re-identifications of IP addresses to online users and households. This work provides a foundation for several new research directions, including the development of methods for learning identity and additional information across disparate datasets, as well as a foundation for methods that enable data holders to share information with guarantees of anonymity.

Keywords: Privacy, Anonymity, Data Mining, Data Sharing, Distributed Databases, Online Privacy

1 Introduction

As people make their way through their daily life they leave behind fragments of information in various databases (Sweeney, 2001). Individuals do not always have control over whether or not their data is collected and, in some instances, they may not even be aware they are shedding any information. For example, images of an individual’s automobile are recorded on different highway video cameras; the IP address of a personal computer is logged at multiple websites; and, a patient’s DNA can be sequenced and recorded in numerous hospital databases. In our data driven society, there is an ever-increasing demand for the incorporation of new technologies to gather data on people for a variety of worthwhile endeavors. Concurrently, data collections have become commodities that can be shared, licensed, or sold for profit in many different communities. This is possible because both data subjects and data collectors consider these fragments innocuous. They often harbor the belief that their pieces of data are isolated and no one could systematically relate identity to them without a central registry to query.

This research demonstrates that not only are such beliefs of anonymity false, but simple learning algorithms can automate the process of linking, or re-identifying identities back to their seemingly anonymous data. The re-identification techniques presented in this paper are a derivation of general learning methods that can be used for such activities as data mining or surveillance, but at the same time they reveal serious privacy risks inherent in current data sharing practices. The ability not only to track people, but to re-identify in the process poses risks to both the individuals and the data holders originally provided with the information. What may be perceived as an abuse of confidential information can result in a loss of trust from consumers or the general public.

Historically, the concept of re-identification, and the development of methods for such a task, was affiliated with the release of data from a single institution or collection (De Waal & Willenborg, 1996; Winkler, 1999). It was believed that a data collection in which each piece of data related to a person could be shared somewhat freely, provided none of the features of the data included explicit identifiers, such as name, address, or Social Security number. However, it was made evident that these collections of “de-identified” data can often be linked to other collections that do include explicit identifiers to re-identify people by name. Fields appearing in both de-identified and identified tables link the two, thereby relating names to the subjects of the de-identified data. For example, {date of birth, gender, ZIP}, which commonly appeared in both de-identified and identified data, uniquely identified 87% of the U.S. population (Sweeney, 2000).

In this paper, we make an extension to trail re-identification, which considers how re-identification can occur via the pattern of locations people visit. The main premise of the trail re-identification model is based upon the observation that people visit different sets of locations where they leave behind similar pieces of de-identified information. The de-identified data may consist of only one or very few fields. Each location visited collects and, subsequently, shares de-identified data on people who visited their location. In some cases, a location also collects and shares, in a different release of data, explicitly identified data, thereby naming some people. When multiple locations share collected data, this allows for trails to be constructed, where a trail is a characterization of the locations that an individual visited. Similar patterns in the trails of de-identified and identified data can be used to link the two.

For a simple scenario, consider the online consumer who leaves the IP address of his computer in access logs at each website visited. At some websites, the consumer may also provide explicitly identifying information; for example, his name and address are provided to complete a purchase. Separately, these websites can share logs containing the IP addresses of those who visited their sites. As e-businesses, these websites can also share explicitly identified data such as customer lists, which typically includes the name and address (e.g. residential, email, etc.) of those who made purchases. By examining the trails of which IP addresses appeared at which locations in the de-identified data and matching those visit patterns to which customers appeared in the identified customer lists, IP addresses can be related to names and addresses. These re-identifications can then be used to identify visits to locations in which the consumer did not make purchases.

From a technical standpoint, this paper introduces a formal model of the general trail re-identification problem and several of its specific variants. It generalizes and extends prior work on genomic data re-identification (Malin & Sweeney, 2001, 2004). Three trail re-identification algorithms are provided, as well as a demonstration of their application on real-world datasets. However, this paper also addresses issues with respect to the impending social atmosphere. Specifically, we consider the erosion of privacy in the online environment. We stress that the threat of trail re-identification is exacerbated by the movement of the online consumer from dial-up modem to broadband Internet connectivity, where always-on connectivity is a main concern.

The remainder of this paper is organized as follows. In the following section the basic terminology, definitions, and data structures for trails are introduced. This section culminates with a formal definition of the trail re-identification problem. In Section 3, three simple algorithms that solve variants of the trail re-identification problem are presented. In Section 4, the algorithms are applied in a set of re-identification experiments with a real-world dataset of online webusers. Following the experimental analysis, in Section 5 we discuss this research in light of recent findings about user behavior in an online environment. In addition, we consider how related work in data re-identification may be adapted to develop more robust trail re-identification methods. Finally, in Section 6, we conclude with a discussion on fruitful directions for computer science research in trail re-identification and the preservation of anonymity.

2 TERMINOLOGY AND DEFINITIONS

In this section we define the fundamental terms and definitions necessary for a formal characterization of data types and trails. In addition, we describe different tactics by which data can be released, as well as the different kinds of trails that result. The section concludes with a formal definition of the trail re-identification problem.

The basics elements are derived from relational database theory. The term *data* refers to information held by a location that collects information on visiting entities. For simplicity, we consider a table to be organized as a set of rows and columns. Each column is referred to as an attribute, which is a semantic category of information that refers to people, machines or other entities that visited the location. The table itself is formally defined as $\tau(A_1, A_2, \dots, A_p)$, where the set of attributes is $A = \{A_1, A_2, \dots, A_p\}$. In the table, each row is a tuple and is specific to a person, machine, or other entity. Each tuple t is defined as $\{[a_1, \dots, a_p]\}$ and represents the sequence of values, $a_1 \in A_1, \dots, a_p \in A_p$. We refer to the size of the table as $|\tau|$, which is the number of tuples.

In the model considered for this research, a particular data-collecting location releases a two-table vertical partitioning of its data, such that one table contains explicitly identified data and the other table is devoid of identified data (i.e. de-identified data). The properties of the partitioned release are formalized in Definition 2.1.

Definition 2.1. (De-identified and Identified Tables). Let $\tau(A_1, A_2, \dots, A_n)$ be a table maintained by a data-collecting location. We define τ^+ as the *identified* subtable of τ , with attributes $A^+ \subseteq A$, where A^+ includes explicit identifying attributes, such as name or address, or attributes linkable to an explicit identifier.¹ Similarly, we define τ^- as the *de-identified* subtable of τ , with attributes $A^- \subseteq A$, such that A^- is devoid of all explicitly identifying attributes, as well as attributes which are directly linkable to an explicit identifier. More specifically, there exists no function which permits the mapping of tuples defined over A^- to tuples defined over A^+ . We refer to the identity or de-identity of a tuple x as *identity*(τ, x) and *deidentity*(τ, x), respectively.

There need not be any relationship between the attribute sets A^- and A^+ and for simplicity this research is situated in an environment where no relationships are known. Furthermore, in a vertical partitioning the order in which tuples appear in the de-identified and identified tables is not necessarily maintained across the tables. Figure 1 illustrates released partitions from website data. Consumer demographics are reported in the identified table τ^+ . IP addresses are reported in the de-identified table, τ^- .

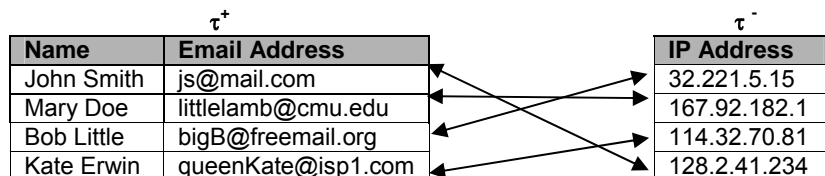


Figure 1. Vertical partitioning by a website of collected data into an identified table τ^+ and a de-identified table τ^- containing IP addresses. There is no reason that the ordering of the tuples in the two tables must be equal, so the arrows are shown to represent the truth of the relationships between tuples.

Notice Definition 2.1 does not require all the tuples in τ to be contained in the identified table τ^+ or in the de-identified table τ^- . However, there do exist certain constraints which are assumed. Specific relationships concerning the number and containment of tuples provided in released tables are discussed in the next section.

2.1 Data Release Tactics

Here we define two tactics by which a location can separate identified from de-identified data prior to sharing. Briefly, in the first tactic, which is called unreserved sharing, the location captures and releases both de-identified and identified data for each visiting entity. Alternatively, the second tactic is a reserved sharing scenario, where the location releases only a subset of one type of data, either the identified or the de-identified. For example, a location may observe the name “John Smith” and the IP address “128.2.41.234”, but the location refuses to release the IP address for this entity. Precise descriptions of these two kinds of releases are provided after several assumptions inherent to this work follow.

¹ For instance, if a dataset contains demographic attributes {date of birth, gender, zip code}, combinations of values over these attributes can uniquely represent an entity and can be directly linkable to public records such as voter registration databases which contain the same fields.

Assumption 2.1. (Truthfulness) Each data-collecting location l releases data collected at l and from no external source. Furthermore, all data released from l is truthful, such that each piece of data correctly represents an entity which visited l .

Assumption 2.2. (Uniqueness of Tuples) In a location’s de-identified and identified tables, each tuple is unique, such that each no two tuples are duplicates.

It follows from uniqueness of tuples assumption that both de-identified and identified tables represent a distinct set of references to people, machines, or other entities that have visited a location, but not necessarily the frequency of visits. As a result, these references narrowly relate to a person, machine, household or other entity to be identified.

Assumption 2.3. (Traceable) Data used to create tracks are traceable. By traceable it is meant there exists, a known non-random relationship between the data an individual’s leaves at different locations.

For an example of traceable with respect to IP addresses, we specify that the same IP address tracked across multiple locations belongs to the same user or set of users.

With our assumptions specified, we continue with our description of release tactics. The first release tactic is referred to as *unreserved* sharing. Definition 2.2 presents a definition for released data that adheres to an unreserved property. When this tactic is employed, only tuples present in the de-identified table have corresponding tuples in the identified table, and vice versa.

Definition 2.2. (Unreserved) Let the table τ be vertically partitioned by the functions V_{id} and V_{de} such that $V_{id}: \tau \rightarrow \tau^+$ and $V_{de}: \tau \rightarrow \tau^-$, where τ^- is the de-identified table, and τ^+ is the identified table. The tables τ^- and τ^+ are *unreserved* if and only if

- (1) $\forall t_{id} \in \tau^+ \exists t_{de} \in \tau^-: V_{id}^{-1}(t_{id}) = V_{de}^{-1}(t_{de});$
- (2) $\forall t_{de} \in \tau^- \exists t_{id} \in \tau^+: V_{id}^{-1}(t_{id}) = V_{de}^{-1}(t_{de});$ and
- (3) $|\tau^+| = |\tau^-|,$

where V^{-1} is the inverse of V .

In releases that adhere to the unreserved property, every tuple from the data-collecting location present in the de-identified table is also present in the identified table, and vice versa. Figure 1 depicts an unreserved release resulting from vertical partitioning of the data into a de-identified table (τ^-) and an identified table (τ^+). Each released tuple has values in both tables.

Nonetheless, releases that are unreserved are not always practical. In some situations, a location may not have collected both identified and de-identified data on all visitors or may not want to share all collected information. In these cases, either the de-identified table or the identified table is incomplete, providing a release that is reserved. The reserved property is more formally defined in Definition 2.3.

τ^+		τ^-
Name	Email Address	IP Address
John Smith	js@mail.com	32.221.5.15
Mary Doe	littlelamb@cmu.edu	167.92.182.1
		114.32.70.81
		128.2.41.234

Figure 2. Vertical partitioning of a website’s collected data into an identified table (τ^+) and a de-identified table (τ^-) containing IP addresses. The lack of identities for two of the IP addresses satisfies the reserved property.

Definition 2.3. (Reserved) Let the table τ be vertically partitioned by V_{id} and V_{de} such that $V_{id}: \tau \rightarrow \tau^+$ and $V_{de}: \tau \rightarrow \tau^-$, where τ^- is the de-identified table, and τ^+ is the identified table. The tables τ^- and τ^+ are *reserved* if either

- (1) $\forall t_{id} \in \tau^+ \exists t_{de} \in \tau^-: V_{id}^{-1}(t_{id}) = V_{de}^{-1}(t_{de});$ or,
- (2) $\forall t_{de} \in \tau^- \exists t_{id} \in \tau^+: V_{id}^{-1}(t_{id}) = V_{de}^{-1}(t_{de}).$

When condition (1) holds, we say τ^+ is reserved to table τ^- ; and vice versa if condition (2) holds.

The tuples of Figure 2 depict a release in which τ^+ is reserved to τ^- . With reference to the true mappings of IP addresses to identity in Figure 1, IP addresses for “Bob Little” and “Kate Erwin” appear in τ^- , but there is no identified information available on them in the released tables.

2.2 Restructured Releases

Given a set of locations, where all locations share releases in either an unreserved or a reserved manner, visits across locations can be tracked by observing which locations reported which visits. These observations are made explicit by constructing a matrix of shared de-identified data and a matrix of shared identified data. These matrices are termed the de-identified track and an identified track, respectively and are formally stated in Definition 2.4.

Definition 2.4. (De-identified and Identified Tracks) Let L be the set of locations sharing their identified tables, τ_i^+ , with attributes A^+ and de-identified tables, τ_i^- , with attributes A^- , where $i \in L$. Let B be a vector containing the members of L , and T^+ be the set of all $\{\tau_i^+\}$ and T^- be the set of all $\{\tau_i^-\}$ for each $i \in L$. Either: (1) τ_i^+ and τ_i^- are unreserved; (2) τ_i^+ is reserved to τ_i^- for each $i \in L$; or, (3) τ_i^- is reserved to τ_i^+ for each $i \in L$.² The *de-identified track*, N , is a matrix having $|A^-| + |L|$ columns. The contents of N are the same as those realized by $FillTrack(N, A^-, T^-)$. Similarly, the *identified track*, P , is a matrix having $|A^+| + |L|$ columns and the contents of P are the same as those realized by $FillTrack(P, A^+, T^+)$ in Figure 3. The number of rows in N and P are $|\cup_{i \in L} \tau_i^-|$ and $|\cup_{i \in L} \tau_i^+|$, respectively.

Less formally, a de-identified track (and an identified track) is a large matrix where each row contains information about a visit and lists the locations in which that visit was reported. The first group of columns in the track is the information collected about a subject on the subject’s visit to a location. The second group of columns is a list of locations. Values associated with these columns correspond to the presence or absence of an entity at a location. For this work, we consider the case when the values of location-based attributes are Boolean, such that a value of 1 specifies an entity’s presence, otherwise a value of 0.

```

FillTrack(Track T, Attributes A, Tables  $\{\tau_i\}$ )


---


Steps:
  Let each cell in T be initialized to 0
  for each location  $l \in L$ :
    for each tuple  $t_i \in \tau_i$ 
      let  $b$  be the index of  $l$  in B
      if there does not exist  $T_j[1, \dots, |A|] \equiv t_i$ , where  $j=1, \dots, |T|$ 
        Let  $k$  be the first unused row in N // has all 0's
         $N_k[1, \dots, |A|] = t_i$  and  $N_k[|A|+b] = 1$ 
      else
         $N_k[|A|+b] = 1$  // another location found

```

Figure 3. Pseudocode for FillTrack, a method for converting a set of tables into a matrix representation of location visit patterns.

In an unreserved release, the identified track (P) and the de-identified track (N) are unreserved. If the tables that construct N are each reserved to the tables that constitute P , track N is reserved to P . Likewise, if the tables that construct P are each reserved to the tables that constitute N , track P is reserved to N . Figure 5 depicts the identified track P and de-identified track N for releases which are unreserved.

2.3 Data Trails

In both de-identified track and identified tracks, the rightmost columns are associated with locations. The vectors of binary values associated with those columns are “trails.” They depict the location where the person, machine, or entity that is the subject of the visits has been recorded. Trails are described more formally in Definition 2.5.

² Note, the case where neither table is reserved to the other can exist. However, while an important case, it is considered beyond the scope of this research paper.

Definition 2.5. (Trail) Let L be the set of locations that share their identified (or de-identified) tables in track T . The shared tables are over the attributes A . A *trail* for subject j is the vector $T_j[|A|+1, \dots, |A|+|L|]$. For convenience, $T_j[|A|+1, \dots, |A|+|L|]$ is written $trail(T, j)$. In addition $trail(T, j, l)$ corresponds to the value for the l^h location.

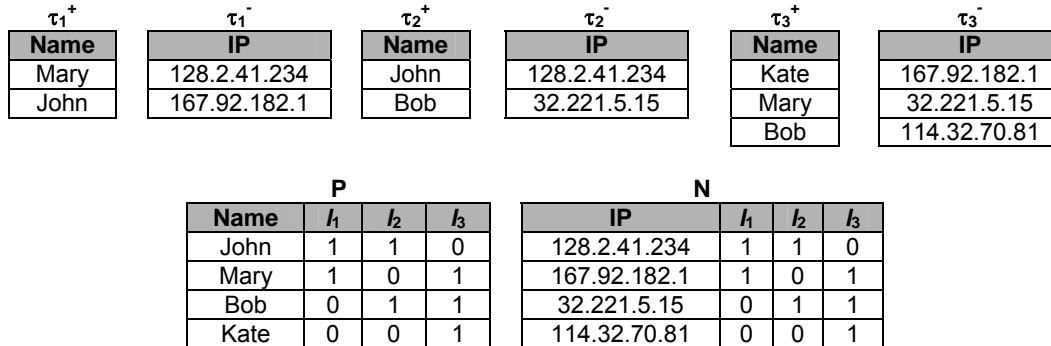


Figure 4. top) The three sets of releases adhere to the unreserved property. bottom) De-identified track N and identified track P are both unreserved.

When we have complete confidence in all of the Boolean values of a trail, we call it a *complete trail* as formalized in Definition 2.6. The releases in Figure 4 adhere to the unreserved property and the resulting trails in both P and N are “complete trails”. Given the identified track P in Figure 4 [1,1,0] is a complete trail for “John” and his de-identified counterpart “128.2.41.234”.

Definition 2.6. (Complete Trail) Let L be the set of locations that share their identified (or de-identified) tables in track T such that the shared tables from each location $l \in L$ are unreserved. A *complete trail* is a trail in T . In a complete trail, values represent the unambiguous presence or absence of a subject at a location such that 0 signifies the subject of the trail did not visit the location and 1 signifies the subject of the trail definitely visited the location.

When de-identified and identified tracks are constructed from a release that adheres to the reserved property, as in Figure 5 then the trails in the reserved track are termed *incomplete trails*. Definition 2.7 presents the concept of an incomplete trail.

Definition 2.7. (Incomplete Trail) Let L be the set of locations that share their identified and de-identified tables in tracks X and Y , such that X is the reserved track of Y for all data holders $l \in L$. An *incomplete trail* is a trail in X . In an incomplete trail, a value of 1 represents the definite presence of the subject at a location and a value of 0 suggests ambiguity. The subject may or may not have visited the location.

Figure 5 depicts the identified track P and de-identified track N for releases which adhere to the reserved property. The identified names tables are reserved to the tables of IP address. Under these conditions track P, which results from the name tables consists of incomplete trails, is reserved to N, which consists of complete trails.

An incomplete trail can match ambiguously to several complete trails. This notion of containment forms the basis for *subtrails* and *supertrails*. See Definition 2.8 and Example 2.8.

Definition 2.8. (Subtrails / Supertrails) Let L be a set of locations that share their identified and de-identified tables in tracks T_1 and T_2 , such that T_1 is reserved to T_2 . The shared tables are over the attributes A . Let x be a trail from T_1 and y be a trail from T_2 . x is a *subtrail* of y (written $x \leq y$) and y is a *supertrail* of x (written $y \geq x$) if and only if: $trail(T_1, x, l) \leq trail(T_2, x, l)$ for all location based attributes l .

For example, in Figure 5 the identified trails [1,0,0], [0,1,0], and [1,1,0] are all subtrails of the de-identified trail [1,1,0]. Similarly, the de-identified trails [1,1,0] and [0,1,1] are supertrails of the identified trail [0,1,0].

With respect to tracks, recall the properties of unreserved and reserved. If tracks N and P are constructed from unreserved releases, then for any particular entity x in the tracks, $trail(N, x, l)$ must equal $trail(P, x, l)$ for all locations l . Furthermore, if N and P are constructed from releases that are reserved, then for any particular entity x in the tracks, $trail(N, x, l)$ must be $\leq trail(P, x, l)$ for all locations d if N is reserved to

P. N consists of incomplete trails and P consists of complete trails. The converse is true if P is reserved to N.

τ_1^+	
Name	
Mary	
John	

τ_1^-	
IP	
128.2.41.234	
167.92.182.1	

τ_2^+	
Name	
John	
Bob	

τ_2^-	
IP	
128.2.41.234	
32.221.5.15	

τ_3^+	
Name	
Kate	

τ_3^-	
IP	
167.92.182.1	
32.221.5.15	
114.32.70.81	

P			
Name	l_1	l_2	l_3
John	1	1	0
Mary	1	0	0
Bob	0	1	0
Kate	0	0	1

N			
IP	l_1	l_2	l_3
128.2.41.234	1	1	0
167.92.182.1	1	0	1
32.221.5.15	0	1	1
114.32.70.81	0	0	1

Figure 5. *top*) The three sets of releases adhere to the reserved property, where a minimum of one identified table (τ_i^+) is reserved to its de-identified counterpart (τ_i^-). *bottom*) De-identified track N is reserved to identified track P, which are constructed from the releases of three locations $l_1, l_2,$ and l_3 .

Tracks A and B are one-to-one if for every entity x represented by a trail in track A there exists only one trail in track B that correctly corresponds to x . In other words, one-to-one means that each IP address is used by one user only. Tracks A and B are one-to-many if every trail in A may correctly be linked to one or more trails in track B. The one-to-many scenario implies that either 1) multiple people share the same IP address and don't use any other IP addresses or 2) one user uses multiple IP address exclusively.

Now that de-identified and identified tracks are defined and how complete and incomplete trails relate to these tracks, the trail re-identification problem can be presented as in Definition 2.9.

Definition 2.9. (Trail Re-identification Problem) Let N and P be de-identified and identified tracks over the attributes A^- and A^+ , respectively. Let there exist a function $f: A \rightarrow B$, where $A \in \{N, P\}$ and $B = \{N, P\} - \{A\}$. A trail re-identification results for a subject s when there exists an i , such that $f(A_s[1, \dots, |A^+|]) = B_i[1, \dots, |A^+|]$. The goal is to determine a function f which maximizes the number of correct re-identifications and minimizes the number of false re-identifications.

For the de-identified track N and identified track P in both Figure 4 and Figure 5 the same function f can reveal re-identifications. The mappings $f(["John"]) = ["128.2.41.234"]$, $f(["Mary"]) = ["167.92.182.10"]$, $f(["Bob"]) = ["32.221.5.15"]$, and $f(["Kate"]) = ["314.32.70.81"]$ are all correct trail re-identifications. The difficulty is in the discovery of such a function, which will be discussed below.

This section precisely described how people, machines, and other entities leave information behind at visited locations, how that information can be shared resulting in trails, and how those trails can pose a trail re-identification problem. In the next section, three novel algorithms for performing trail re-identifications are presented.

3 REIDIT ALGORITHMS

Given L , the set of data-collecting locations whose shared tables result in de-identified track N and identified track P over the attributes A^- and A^+ , respectively, algorithms that exploit the uniqueness of trails in N and P can be written to perform trail re-identifications. The three algorithms presented in this section are variants of this approach. Collectively, they are termed Re-identification of Data in Trails (REIDIT).

3.1 REIDIT-Complete

The first algorithm is named REIDIT-C. REIDIT-C assumes that both N and P are unreserved, and therefore, REIDIT-C only works on complete trails. It performs exact match on the trails of N and P as follows. For every trail in N, REIDIT-C determines if there exists one and only one trail in P such that the trails are equal. When there is an exact and unique match, then $trail(N, n)$ is re-identified to explicitly identifying information in P. If $trail(N, n)$ is equal to $trail(P, p)$, and there exists another $trail(P, p')$ also equal to $trail(N, n)$, then there is ambiguity and no re-identification can occur. The formalization of REIDIT-C is provided in Figure 6.

Complexity. First, the outer loop iterates over all of the records in N , which is $|N|$ iterations. Second, for each iteration in N , the algorithm iterates a maximum of $|P|$ times. This provides $O(|N|*|P|)$ or $O(|N|^2)$ because $|N|=|P|$. Yet, this analysis is an artifact of the way in which the pseudocode is written. An actual implementation can be written such that each set of trails are sorted and then compared, resulting in $O(|N|\log|N|)$.

Algorithm: REIDIT-C(N, P)

Input: De-identified and Identified Tracks N and P over attributes A^- and A^+ , respectively, for the same data-collecting locations.

Output: Set of re-identifications R

Assumes: 1) N and P are unreserved, 2) N and P are one-to-one

Steps:

```

1   let  $R = \emptyset$ 
2   for  $x=1$  to  $|N|$ 
3     let  $M = \emptyset$ 
4     for  $y= 1$  to  $|P|$ 
5       if there exists one and only one  $trail(N, n_x) \equiv trail(P, p_y)$ 
6          $R = R \cup \langle identity(P, p_y), deidentity(N, n_x) \rangle$  //  $\langle n_x, p_y \rangle$  is a linked identity-deidentity pair
7   return  $R$ 

```

Figure 6. Pseudocode for the REIDIT-C algorithm.

Theorem 3.1 *Trail re-identifications from REIDIT-C are correct, such that no false re-identifications (i.e. the linkage of two pieces of data which do not correspond to the same entity) are made.*

PROOF: First, recall the underlying assumption of the unreserved-release model: tuples of both tracks N and P consist only of complete trails. Thus, at location l , a visit from an entity must be recorded in both T_l^- and T_l^+ . Since this holds true for every location, for each $trail(N, n)$, there must exist at minimum one equivalent $trail(P, p)$. If there exists more than one equivalent trail in P for $trail(N, n)$, then multiple trails will be recognized and the singleton requirement will not be satisfied. No re-identification will be recorded. \square

3.2 REIDIT-Incomplete

The second algorithm is named REIDIT-I. It performs subtrail/supertrail matching on the trails of N and P . REIDIT-I assumes one of the following as true: either N is reserved to P or P is reserved to N .

When an incomplete trail can be matched to a single complete trail, a trail re-identification occurs. Unlike REIDIT-C, however, equality cannot be relied upon for matching trails. Instead, containment of subtrails by supertrails is used. For each trail in the track containing incomplete trails, the set of its supertrails from the track containing complete trails are found. If there is only one supertrail, then a correct trail re-identification has occurred. The re-identified trails from N and from P are removed. Processing continues until no more re-identifications can be made because one of two conditions is satisfied: either (1) N or P has no more trails to process; or, (2) there are no distinct re-identifications possible in the current iteration. This simple view of the algorithm is presented in Figure 7.

Complexity. The complexity of the algorithm is best understood by studying an alternative representation of the one in the steps shown in Figure 7 provided above. Without loss of generality, let X , Y be N , P , respectively. The re-identification process can be made more efficient by precomputation of an adjacency matrix Z , where $Z[n, p] = 1$ if $trail(N, n) \leq trail(P, p)$, and a vector S of length $|N|$, where each cell $S[n]$ is the rowsum of the n^{th} row of Z . The precomputation step is completed in $O(|N|*|P|)$. The re-identification process proceeds as follows. The vector S is sequentially scanned. When $S[n] = 1$, the n^{th} row of the Z matrix is scanned until $Z[n, p]=1$ is found. These coordinates reveal a re-identification, so the n^{th} entry of N is re-identified by the p^{th} entry of P . Next, each cell $S[x]$ is subtracted by $Z[x, p]$, and if $Z[x, p]=1$ it is set equal to 0. This scanning process is continued until no cell in S is equal to 1. In a worst-case scenario, each scan of S yields one re-identification, thus taking $|N|$ iterations. During each iteration, a sequential scan of the S vector takes place in $O(|N|)$ time. Once a $S[x]=1$ is found, a scan of one row of the Z matrix occurs in $O(|P|)$ steps. When cell $Z[x, y]$ with value 1 is found, the found column in Z and the S vector are updated with a scan taking $O(|N|)$ steps. Since, in worst case there is only one re-identification per do-while iteration, this process only occurs once per iteration. Thus, the total number of

steps is approximately $|N|^*(2*|N|+|P|)$, which is approximately $O(|N|^2 + |P|^*|N|)$. Therefore, the order of complexity will be $O(matrix\ setup) + O(matrix\ scanning)$. Since $|P| \geq |N|$, complexity is $O(|N|^*|P|)$.

Algorithm: REIDIT-I (N, P)

Input: From de-identified and Identified tracks N and P over attributes A^- and A^+ , respectively, for the same data-collecting locations, X is the reserved table of N or P and Y is the other table. For simplicity, let X be N and Y be P.

Output: Set of re-identifications R

Assumes: 1) X has incomplete trails and Y has complete trails, 2) X and Y are one-to-one, and 3) X is reserved to Y

Steps:

```

1   let R = ∅
2   Do
3       PreviousREID = |R|
4       for m=1 to |X|
5           If there exists one and only one trail(X,xm) ≤ trail(Y,y)
6               R = R ∪ ⟨identity(X, xm), deidentity(Y,y)⟩ //⟨xm,y⟩ is a linked identity-deidentity pair
7               X = X - xm, Y = Y - y //remove x and y from further consideration
8   while PreviousREID ≠ |R|
9   return R

```

Figure 7. Pseudocode for the REIDIT-I algorithm.

Theorem 3.2 Trail re-identifications from REIDIT-I are correct, such that no false re-identifications are made.

PROOF: Without loss of generality, let X, Y be N, P, respectively. N has incomplete trails and P has complete trails. From the definition of incomplete trails, all 1's are correct, so it must be true that for an arbitrary trail in N, there must exist a non-null set of supertrails whose identifying information appears in $S[n]$ for $trail(N,n)$. If $S[n]$ is equal to 1, then there exists only one complete supertrail that could be reconstructed for $trail(N,n)$ through the replacement of 0's with 1's. Therefore $trail(N,n)$ is re-identified in by the column in Z, where $S[n,m]=1$. In the event when $S[n] > 1$, then the algorithm can still converge to a correct re-identification as follows. Let $S[n]$ equal k. When a re-identification is made for a trail other than $trail(N,n)$, then $S[n]$ decreases by 1. Since it is already known that $S[n]$ has a minimum of 0, if $S[n]-1$ re-identifications are made for trails of N, excluding $trail(N,n)$, each with a member contributing to $S[n]$, then the remaining member of $S[n]$ must re-identify $trail(N,n)$. □

While the version of REIDIT-I presented in Figure 7 is correct in the re-identifications that it discovers, it should be noted that a greater number of re-identifications can be discovered if the special case $|N| = |P|$ exists. When N is reserved to P, it is guaranteed that for each tuple $n \in N$, there exists a tuple $p \in P$ that re-identifies n. Yet, it is not true that every tuple in P can be re-identified by a tuple in N. So, if a tuple in P is found to have only one subtrail in N, this may not be a correct re-identification. This is why the track that is reserved to the other is put on the outer for loop in REIDIT-I.

However, when $|N|=|P|$, the every trail in P has a corresponding trail in N. Thus, after the re-identifications from N to P are discovered and every remaining unidentified $S[n] > 1$, there can exist columnsums of size 1 will also reveal correct re-identifications from P to N. In this specific case, lines 10-13 of Figure 8 should be inserted between lines 7 and 8 of Figure 7. Basically, we add a second while loop with the tracks exchanged. This allows for us to search for complete trails which map to a single incomplete trail.

```

10      If |X| = |Y|
11          for n=1 to |Y|
12              If there exists one and only one trail(X,x) ≤ trail(Y,yn)
13                  R = R ∪ ⟨identity(X, x), deidentity(Y,yn)⟩ //⟨x,yn⟩ is a linked identity-deidentity pair

```

Figure 8. Pseudocode for REIDIT-I in the special case when $|X| = |Y|$.

Incorporation of the second loop does not have much influence on the overall complexity of the REIDIT-I algorithm. Complexity will remain $O(|N|^*|P|)$, but can now be simplified to $O(|N|^2)$, since $|N|=|P|$.

3.3 REIDIT-Multiple

The third algorithm is named REIDIT-M. It allows multiple references in P to be related to only one reference in N, or vice versa. For example, multiple individuals in a shared setting, such as a household, can use the same computer. Online purchasers, in this case, would have multiple identities related to the same IP address. The reverse is also possible. One person could use more than one computer and therefore one reference in P would relate to multiple references in N. REIDIT-M addresses collocation issues, such as these. REIDIT-M assumes either N is reserved to P or P is reserved to N.

Unlike REIDIT-I, the REIDIT-M algorithm relaxes the assumption that there must be one-to-one relationship between trails. If an incomplete trail is a subtrail of only one supertrail, then a re-identification occurs via a linkage between these two trails. Multiple subtrails can map to the same supertrail and permit a re-identification. REIDIT-M is provided in Figure 9.

Algorithm: REIDIT-M (N, P)

Input: From de-identified and Identified Tracks N and P over attributes A^- and A^+ , respectively, for the same data-collecting locations, X is the reserved table of N or P and Y is the other table. For simplicity, let X be N and Y be P.

Output: Set of re-identifications R

Assumes: 1) X has incomplete trails and Y has complete trails. 2) X to Y is one-to-many.

Steps

```

1  Let  $R = \emptyset$ 
2  For  $n=1$  to  $|X|$ 
3      if there exists one and only one  $trail(X, X_n) \leq trail(Y, y)$ 
4           $R = R \cup \langle identity(X, X_n), deidentity(Y, y) \rangle$  //  $\langle X_n, y \rangle$  is a linked identity-deidentity pair
5  return R
    
```

Figure 9. Pseudocode for the REIDIT-M algorithm.

Complexity. Let X be N and Y be P. First, the outer loop iterates over all of the records in N, which is $|N|$ iterations. Second, for each iteration in N, the algorithm iterates a maximum of $|P|$ times. Thus, the algorithm is $O(|N|*|P|)$.

3.4 Upper Bounds

For both REIDIT-C and REIDIT-I, the maximum number of trail re-identifications is dependent on the number of permutations of a binary string. Given an identified track P, containing references to subjects and the locations visited, and a set of data-collecting locations L, if $|P| \leq |L|$, then the maximum number of trail re-identifications is bounded by $|P|$, the number of subjects. This implicates that all trails may be re-identified. When $|P| > |L|$, the maximum number of trail re-identifications is bounded by the number of locations in the exponential manner $2^{|L|}-1$. When $|P| > 2^{|L|}$, it will be impossible to re-identify all trails. In contrast, for REIDIT-M, the number of re-identifications is independent of the number of data-collecting locations, because it is possible for multiple trails in P to be mapped to a single unidentified trail. As such, the maximum number of re-identifications is $|P|$.

4 RE-IDENTIFICATION EXPERIMENTS

Though in theory the re-identification limits of REIDIT scale exponentially, this is not guaranteed in the real world. One of the main reasons is that entities are not random agents who generate binary strings with uniform probabilities. On the contrary, much research in e-commerce and web personalization suggests that there are trends in the way that users access websites. To analyze some of the intricacies of real world data with respect to the REIDIT algorithms, we study a real world dataset consisting of household online usage behavior. The dataset was compiled by the Homenet project (Kraut, 2002) at Carnegie Mellon University, who provide families in the Pittsburgh area with internet service in exchange for the monitoring and recording of the families' online services and transactions. We used URL access data collected over a two-month period that included 86 households and 144 individuals. Each individual/household was issued a unique login and password for fine-grained monitoring. Overall, approximately 5000 distinct website domains and 66,000 distinct pages were accessed. Some simple summary statistics follow (though the reader should keep in mind the power-law skew of the website visit

distribution): the mean for number of websites visited per user was about 100, with a standard deviation of 120 sites.

The following experiments analyze the re-identifiability of the online browsing patterns in the Homenet dataset. The first set of experiments analyzes the re-identifiability of the online browsing and actual purchasing behavior of the individuals and households in the Homenet study. The second set of experiments considers the affect of location population on the ability to re-identify a population of online users.

4.1 Re-Identifiability of Observed Browsing Behavior

For the following set of experiments we reconstructed purchase data and weblogs for websites accessed by this population. The URL data was manually labeled as “purchase made” or “purchase not made” as inferred from the accessed page. For example, a purchase confirmation URL at Greyhound.com was labeled as a purchase, while the frontpage of the website was labeled as not being a purchase. It was determined that purchases were made at 24 distinct websites, including Amazon.com, Ticketmaster.com, and Hotwire.com. We make the assumption that websites collect two types of data: 1) identifying information, such as name or address on the purchaser at the time of purchase; and, 2) the IP address of computers visiting their site on each visit. For the most part, these websites were more popular than the average website, with a mean of approximately 6 websites visited per user, with a standard deviation of about 3. In comparison, the number of websites each user made purchases at was smaller, with a mean of 2 locations visited per user and standard deviation of about 1.

4.1.1 Complete Re-identification

In this experiment, two scenarios were explored, trail re-identifications to online users and trail re-identifications to households. For re-identifications to online users, the attributes released with the de-identified tables were $A_{per} = \{website, purchaser\ IP\ address\}$ and the attributes released with the identified tables were $A_{per}^+ = \{website, name, address\}$ for each targeted website location. De-identified track N_{per} and identified track P_{per} were constructed having 30 rows. The number of locations was 24. For re-identifications to computer households, the attributes released with the de-identified tables were $A_{hou} = \{website, household\ IP\ address\}$ and the attributes released with the identified tables were $A_{hou}^+ = \{website, street\ address\}$ for each targeted website location.

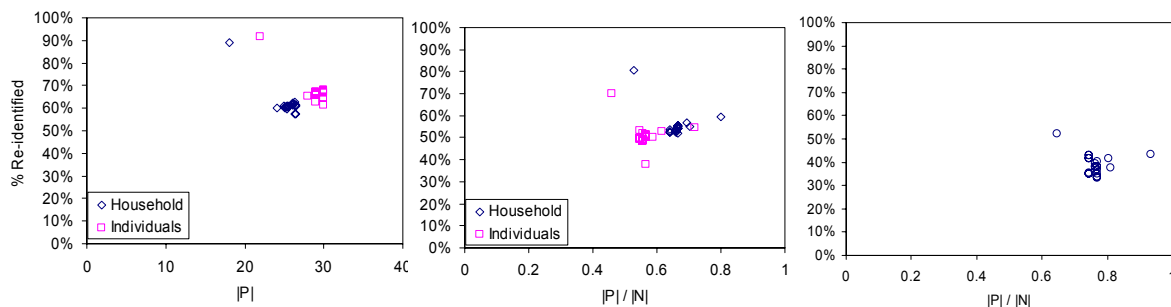


Figure 10. Sensitivity of re-identification to one location left out in data with *left*) an unreserved, *middle*) an reserved, and *right*) a multiple release. The x-axis in the left graph, and the y-axis in the remaining two, are slightly jittered for visual inspection.

De-identified track N_{hou} and identified track P_{hou} were constructed having 26 rows. The number of locations was 24. REIDIT-C performed trail re-identifications on N_{per} and P_{per} and on N_{hou} and P_{hou} ; they all contain complete trails. There were 30 individuals, which made up 26 households, with purchases at a total of 24 websites. Of these trails, 16 IP addresses, approximately 62%, were re-identified to mailing address and 20, approximately 66%, individuals were re-identified. To determine the sensitivity of REIDIT-C to additional withholdings of certain locations, further analysis was conducted with respect to the removal of single location. The experiment was run 24 times, each time leaving out a new location. Trail re-identifications using REIDIT-C was minimally affected. The results are shown in the leftmost graph of Figure 10. The percent re-identified corresponds to the percent of the remaining population after a location was removed. The observed outlier corresponds to a website (Ticketmaster.com) that was

accessed by many purchasers, but played a minimal role in trail re-identification. Removal of this website allowed for around a 25% improvement in trail re-identification. This experiment demonstrates that IP address can be re-identified in some cases, thereby compromising the geographic privacy of the IP address.

4.1.2 Incomplete Re-identification

Using the dataset described above, websites now reported IP addresses for all visitors to their site, regardless of a purchase or not. Again, we explored two scenarios, trail re-identifications to online users and trail re-identifications to households. For re-identifications to online users, the attributes released with the de-identified tables were $A_{per}^- = \{\text{website, individual IP address}\}$ and the attributes released with the identified tables remained $A_{per}^+ = \{\text{website, name, address}\}$ for each targeted website location. De-identified track N_{per} had 53 rows and identified track P_{per} had 30 rows. N_{per} has incomplete trails. P_{per} has complete trails. The number of locations remained 24. For re-identifications to computer households, the attributes released with the de-identified tables were $A_{hou}^- = \{\text{website, household IP address}\}$ and the attributes released with the identified tables remained $A_{hou}^+ = \{\text{website, street address}\}$ for each targeted website location. De-identified track N_{hou} had 39 rows and identified track P_{hou} had 26 rows. N_{hou} has incomplete trails. P_{hou} has complete trails. The number of locations remained 24.

Trail re-identification was performed through REIDIT-I. For this experiment, the 24 websites release IP data corresponding to 39 households and 53 individuals. REIDIT-I re-identified 9 IP addresses, approximately 35%, to households and 15 to individuals, approximately 50%. Sensitivity of REIDIT-I to single locations was analyzed in the same leave one out manner as performed with the previous experiment. The results are provided in Figure 10. One location, Amazon.com, had a significant effect on the ability to re-identify individuals, in that removal of this location decreased the size of the considered population and increased the ability to re-identify IP addresses by about 25%.

4.1.3 Multiple Re-identification

Using the dataset described in section 4.1.2, we acknowledge that a household may have multiple users of a particular computer. In this experiment, each website releases a list of customers who made a purchase at the website, where the list includes the email address, not the mailing address of the purchaser. An IP address of a computer may now relate to multiple email addresses. The attributes released with the de-identified tables were $A^- = \{\text{website, IP address}\}$ and the attributes released with the identified tables were $A^+ = \{\text{website, email address}\}$ for each targeted website location. De-identified track N had 30 rows and identified track P had 23 rows. The number of locations remained 24.

There were 23 households with a single purchasing individual, 2 households with 2 individuals, and 1 household with 3 individuals (i.e. a total of 30 individuals). REIDIT-M achieved trail re-identification for all three full households. REIDIT-I, however, failed to recognize them. In the Homenet dataset, family members visited common sites, which under REIDIT-I remain ambiguous at the individual level, but not for REIDIT-M at the household level. Sensitivity of REIDIT-M to single locations was analyzed as described before and results are shown in Figure 10.

4.2 Sensitivity Analysis of Location Popularity

In the previous set of experiments, we touched upon the sensitivity of a population to re-identification as a function of the locations that were releasing data. In the following analyses, we investigate this concept more specifically by performing a sensitivity analysis of a location's popularity on re-identification susceptibility.

In previous studies, it was found that the popularity of webpages within a particular website varies widely with high skew (Breslau et. al., 1999). The popularity adheres to a power-law function, namely the Zipf distribution. In this type of distribution, given a set of pages and a set of users which visit the pages, a two dimensional log-log plot of page rank (in terms of number of visitors) vs. the actual number of visitors will follow an inverse linear trend. This finding is validated in a number of environments, including traffic over websites and, subsequently, has been employed for the design of more efficient search engines (Brin & Page, 1998).

In a Zipf distribution, the probability of occurrence of an event, f_i , is inversely proportional to the event's rank (as determined by its frequency) r_i ,

$$A * f_i = r_i^{-\alpha} \tag{1}$$

where α is a constant between $[0,1]$ and A is the number of observations. With respect to the re-identification studies of this research, consider an environment where L is a set of locations and A is a population of subjects visiting those locations. The probability that any particular entity visits location $l \in L$ is equal to $r_l^{-\alpha}$, where r_l is the rank of l 's popularity. Subsequently, the number of entities which visit l is $A * r_l^{-\alpha}$. Moreover, the α constant is a term which controls the magnitude of skew, such that when α equals 1, the distribution is a true Zipf and when $\alpha < 1$ the Zipf distribution is said to be in a generalized form. As a result, the log-log plot of "number of visitors" to "location rank" is linear, while the coefficient functions as a dampening factor on the slope of the plotted curve.

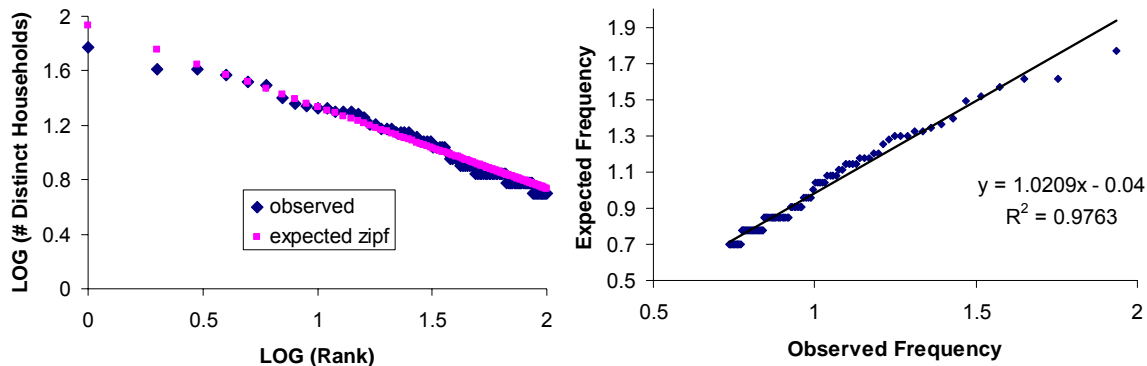


Figure 11. *Left*) log-log (generalized Zipf) distribution of Homenet households accessing the 100 most popular websites. *Right*) Correlation plot of observed and expected log(frequency), or the number of distinct household visits to each location.

For our studies, we consider f_i to be the probability that an individual visits website i and A is the set of households in the Homenet dataset. To determine if the Homenet dataset is representative of a real world environment, we analyzed the traffic at each domain with respect to the number of distinct visitors. The log-log plot of the observed distribution is shown in Figure 11. Similarly, in this plot, we show the expected frequencies for a Zipf distribution with an α of 0.6. A linear fit of observed frequencies to expected frequencies yields a correlation coefficient of ~ 0.98 , so it is apparent that this data trend does hold in the Homenet dataset.

4.2.1 Additive Complete Re-identification

It would be ideal to continue with the labeled dataset from the previous set of experiments. However, to perform a more in-depth analysis of web browsing behavior, we continue with the full 86 households of the Homenet dataset. We make the simplifying assumption that when an individual visits a website, both their IP address and their identifying information is left behind.

With respect to re-identifiability, the Zipf distribution suggests that websites accessed less often should be more useful than others for re-identification using the REIDIT algorithms. To evaluate this belief with REIDIT-C, we analyzed the number of households re-identified as a function of the number of websites that made up a trail. We considered an environment where an increasing number of websites participate in unreserved data sharing (i.e. if IP address is released, then identifying information is released as well). We analyze the effect of the addition of a single website by adding websites in reverse order of their popularity rank. Figure 12 depicts the effect. We find that there is logarithmic growth in the number of re-identifications as a function of increased websites.

Moreover, the plot of re-identified households is actually the same plot as the number of households with non-null trails generated by the websites considered. This effect and the logarithmic growth is expected, since lesser accessed websites are well distributed across much of the population. Eventually, remaining

users are those who tend to visit high traffic sites. Thus, it appears that users who visit low traffic websites and reveal identity are highly susceptible to trail re-identification.

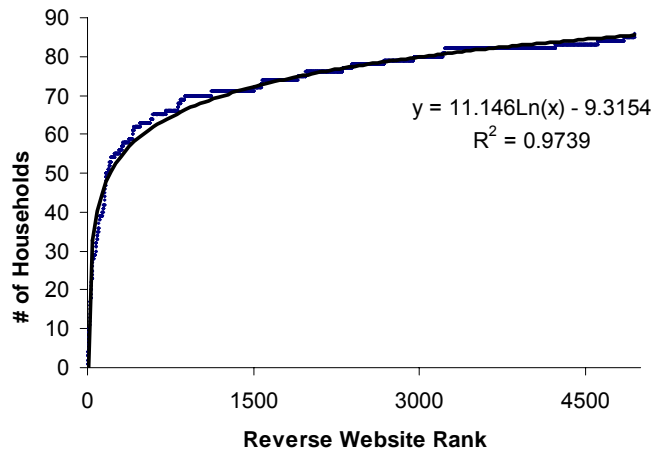


Figure 12. Re-identification of complete trails with number of websites increasing from least-visited to most-visited.

However, what about the users who tend to connect to heavily accessed websites? Can highly traveled websites provide enough variation to re-identify individuals? And if so, how many pages are necessary for the population? To analyze such questions, we flip our analysis via website rank around. For this experiment, we consider the more popular websites. Figure 13 depicts this analysis, where the line labeled “discovered” in this graph represents the number of households with non-null trails made up of information from the considered websites. We find that the actual number of households re-identified is slow to approach the theoretical number of possible re-identifications. Though this suggests users have similar visit patterns over highly as the number of websites contributing to a trail increases, the number of re-identifications also increases and by 20-25 websites, almost all re-identifications are discovered.

4.2.2 Additive Incomplete Re-identification

The complete trails from above were used to generate incomplete trails for analysis of the REIDIT-I algorithm. To do so, we utilize a simple model of how websites create reserved releases (i.e. IP address released, but not identifying information). Each website withholds identifying information on a visiting household with the same probability p . Thus, the track of complete information consists of IP address trails and the track of incomplete information consists of identifying information trails.

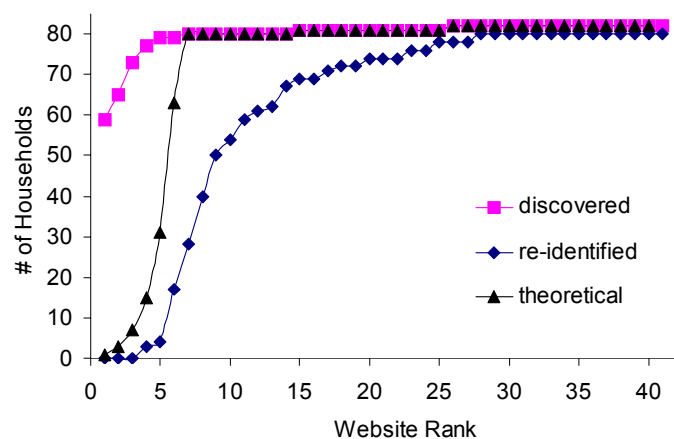


Figure 13. Re-identification of complete trails with number of websites increasing from most-visited to least-visited.

We varied the probability of information being withheld and attempted re-identification with REIDIT-I. Graphs of the results for p equal to 0.1, 0.5, and 0.9 are shown in Figure 14. Each point of a graph depicts the average result for 100 experiments of random information withholding. As the probability of withholding information increases, the probability that an individual will not show up at all (i.e. no trail generated) in the population of incomplete trails. Thus, in the graphs we show three trends. The topmost line represents the number of non-null IP address trails for a given set of websites. The middle line represents the number of non-null identifying information trails. And the bottom line represents the number of IP addresses that were re-identified.

We find that as the amount of information withheld increases, the number of websites necessary to perform re-identification increases as well. This is because as additional information is withheld, the incomplete trail becomes less complex and informative. However, even though trails become less complex, there remains a significant disposition toward re-identification. This is observable even after 50% of a trail is obscured. We find that there is an inverse relationship between the slope of re-identification (as a function of website rank) and the amount of information withheld. Thus, though information withholding may decrease the trail re-identification susceptibility, a costly amount of information must be obscured. This finding may be derivative of using a uniform probability of information withholding. In future studies, we plan to analyze more complex models, which may provide the ability to release more information while preventing trail re-identification.

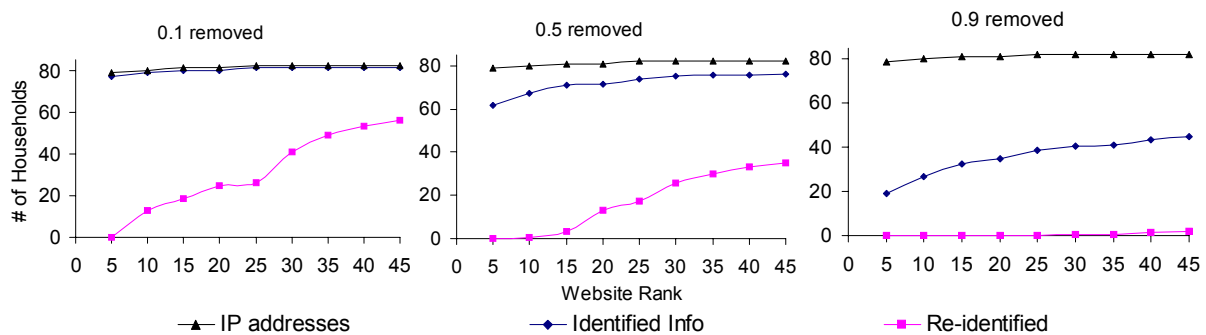


Figure 14. Re-identification of incomplete trails as an increasing amount of identifying information is withheld from the release. Information is removed from the release with a left) 0.1, center) 0.5, and right) 0.9 probability.

4.2.3 Additive Multiple Re-identification

As stated above, there exist many households with multiple users for a single IP address. In this experiment, we allow an IP address to be re-identified to multiple identified individuals. The full Homenet dataset consists of 56, 9, 14, 5, 1, and 1 household(s) of 1, 2, 3, 4, 5, and 6 people respectively.

Initially, the number of people and the number of households re-identified are similar and grow at a similar rate. However, as depicted in Figure 15 after approximately 60 websites, the re-identification of individuals begins to surpass the growth rate for households. From this test, it is apparent that members of the same household visit different sets of webpages. If members of the same household visited the same set of webpages, the growth rate of re-identification for individuals would be steeper than that of the household discovery rate at an earlier point. While this analysis demonstrates that complete trails of IP addresses are complex enough to re-identify multiple individuals, more websites are needed for re-identification of the population than with REIDIT-C. This is not surprising, since REIDIT-M allows searches for unique matches of incomplete trails to complete trails and matching multiple incomplete trails against a complete trail may require additional information.

5 Increasing Risk

One of the major assumptions of this work is in the belief that de-identified data, such as IP addresses, are readily relatable for an individual or household. The correctness and power of the REIDIT algorithms above derive from the ability to trace an individual from one website to another through an IP address. In

today's society, however, it is easy to make an argument against such an assumption. First of all, consider that Internet users subscribe to Internet Service Providers (ISP) that assign IP addresses to users in a dynamic, not static, manner. Each time a user establishes a connection to the internet, his IP address may be different. Thus, if an IP address is owned by a traditional dial-up ISP, then the longer the amount of time an IP address is tracked for, the greater the probability the trail will be composed of multiple unrelated users. This problem is made more daunting by the finding that narrowband dial-up (56kbps or less) users have online sessions of much shorter length than users of broadband connections (Rappoport, Kridel, & Taylor, 2002). Second, as of May 2003, approximately 65% of users within the United States connect to the Internet using narrowband dial-up modems (Bandwidth Report, 2003). Therefore, it appears that the threat of trail re-identification is not really as major as implied, since it is only applicable to data whose collection time ranges over a very brief time period.

Yet, the assumption of static data is not as detrimental as it seems at a first glance. This is due to two aspects, one sociological, the other technological. The previous derives from the migration of Internet users from simple dial-up modems to broadband Internet connections. The latter relates to prior research in probabilistic and computational methods of re-identification that may be adapted to trail re-identification. Not only will both aspects not make trail re-identification of IP addresses more feasible than at the current point in time, but the ability for re-identification will become better with time. Both of these issues we now discuss.

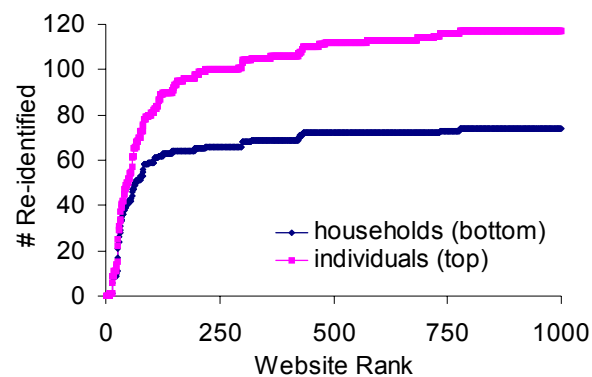


Figure 15. Re-identification of incomplete trails of individuals to complete trails of household IP addresses.

5.1 Broadband Shift

Internet users are migrating to faster Internet connection alternatives, mostly broadband technologies. But users are not merely adopting faster connections, rather, the technology is changing the way that individuals actually use the internet. Over the past several years, the adoption of broadband has been progressing at an annual rate of approximately 0.3 times the current population of broadband users (i.e. 10 users today, but 13 users next year) (Wireline Competition Bureau, 2002). While as of May 2003, 35% of the current population uses broadband, by mid 2004 over 50% of users will connect via broadband. (Rappoport, Kridel, & Taylor, 2002) The implications for trail re-identification are noteworthy.

Broadband users tend to stay continuously connected to the Internet, with the same IP address, for longer periods of time (Rappoport, Kridel, & Taylor, 2002). This shift to broadband supports the tracking of IP addresses and, as this shift continues, trail re-identification will be able to be conducted over data collected over larger time ranges. An even more interesting finding of broadband users is the way they actually visit websites. It has been discovered that broadband users not only visit more websites in comparison to narrowband users, but they participate in more e-commerce (Rappoport, Kridel, & Taylor, 2002). Thus, broadband users not only leave behind their de-identified information at more sites per session, but they leave behind identified information more often. All of these findings imply that as the population continues to adopt broadband, trail re-identification will become more robust. The increased capability will derive from longer time ranges, an increased upper bound for re-identification, and more complex web browsing behavior. The latter of these three will probably play the biggest role in re-identification. As web browsing behavior becomes more complex, the less sparse a trail becomes, and the easier it is for an IP address trail to be re-identified.

5.2 Related Re-identification Research

Trail re-identification is not the first type of re-identification to be studied. Mainly, previous research has been conducted within a variety of communities. Here we consider the three most related: record linkage, data linkage, and data mining. With the quantity of research and rich theory that has already been developed in these areas, it is possible that techniques from any of these fields may be adapted for trail re-identification. Here, we provide a brief overview of related re-identification research.

The problem that record linkage (Bilenko & Mooney, 2003; Elfeky, Verykios, & Elmagarmid, 2003; Newcombe, Kennedy, Axford, & James, 1959; Fellegi & Sunter, 1969; Sarawagi & Bhamidipaty, 2002; Winkler, 1995, 2002) attempts to solve is how to automate the updating of two lists, A and B, or the deduplicating of a single list. The process of record linkage corresponds to building a statistical model to classify pairs from the product space $A \times B \rightarrow \{M, U, C\}$, where M is the set of definite matches, U is the set of definite non-matches, and C is the set of pairs that need clerical review. The goal is to minimize the error in the sets M and U , while minimizing the size of C . It is assumed that there are two files with common variables and that there is typographical error or alternate representation of information (e.g. John Smith vs. Jon Smith) in the files. Initially, the process was not designed for compromising privacy, but rather to relate records of an individual for which minor corruption in one or both of the records has occurred. While the technique does relate the records of a particular subject, for the most part, record linkage has not been associated with associating de-identified data to identified data. Furthermore, in future research it is necessary to consider trail re-identification in more variable environments, such as when a webuser reveals his name differently to various locations or when typographical error exists.

Data linkage differs from record linkage in several fundamental aspects. The most notable difference is that data linkage techniques are specifically designed for re-identification purposes. In addition, the attributes of the two files are not required to be the same. Instead data linkage is concerned with exploiting inferential relations between attributes of the two files. A combination of the values in the attributes is utilized to estimate the uniqueness of an entity's identity in a known population (Sweeney, 2000). Fields appearing in both de-identified and identified tables link the two, thereby relating names to the subjects of the data. For example, $\{date\ of\ birth, gender, ZIP\}$, which commonly appears in both tables, uniquely identifies 87% of the U.S. population. When a de-identified record cannot be uniquely re-identified, the process ceases for the considered record. It appears that the trail re-identification problem is most related to data linkage, where it extends such a procedure into a simultaneous evaluation over a large number of tables.

A third method of re-identification relies on ordered weighted aggregation (OWA) operations (Torra, 2000, 2001, 2004). This approach attempts to re-identify when there are no common attributes between releases. The procedure takes a table of records and performs dimensionality reduction by converting the data vector V of a record $[v_1, v_2, \dots, v_n]$ into a new vector W of several weighted scalars $[w_1, w_2, \dots, w_m]$, where $m < n$ and w_i is a weighted scalar for the i^{th} parameterization of the OWA operator. The goal is to create an ordering of the data using combinations of attributes. Re-identification is then achieved by matching records from disparate tables that have similar weighted W vectors. The technique has been demonstrated to work well for the re-identification of attributes, where the data vectors are the values of an attribute for all records. And while the claim has been made that this technique can re-identify individual records in a table, there is no proof to substantiate this.

6 Conclusions and Future Research

The REIDIT algorithms provide deterministic methods for learning who (by name or explicit identity) has been where. The methodology involves constructing trails across websites from small amounts of seemingly anonymous or innocuous evidence that a person has visited there. Trails are also constructed on places where the person has left explicit information of their presence. Identifying uniqueness and inferences across these two sets of the trails relates information about where the person has been to who they are.

The development of the REIDIT algorithms is important to society simply because people seek safety without unnecessarily relinquishing their privacy. Clearly, the REIDIT algorithms exacerbate privacy concerns. The fact that trail re-identification can be done, as evidenced by the existence of this work,

informs society and data privacy researchers of a real challenge to protecting privacy. However, our research is merely a beginning, and there are several areas of research that must be explored to expand this work. We conclude this paper with a discussion on two main issues for research in trail re-identification: 1) defeating and 2) extending trail re-identification. The challenge is for some researchers to attempt to thwart these approaches by improving data privacy methods, while others try to improve their ability to learn. With an open and aggressive pursuit on both sides, we as computer scientists can best inform society and help play a crucial role in the debates of our time.

6.1 Defeating Trail Re-identification

Currently, there is no documentation on how particular protection schemas might thwart the various trail re-identification methods. However, there are several promising areas of research in privacy protection that may be of utility. Here we address three promising concepts, statistical protection, computational disclosure control, and secure multiparty computation. While each of these communities offers viable models and methods for protecting data, there is no formal modeling of how protecting against trail re-identification will affect data sharing or usefulness of the resulting data.

Statistical-based methods attempt to protect data through techniques based on the following dogma. The receiver of the released data should be able to reconstruct accurate aggregate distributions, while the exact values of a record can not be determined. Many of the established methods in this community are based on the addition of noise, or perturbation, of the records in a released collection (Agrawal & Srikant, 2000; Cox, 1980; Chowdhury et. al., 1999; Duncan & Fienberg, 1997n). A problem with statistical-based methods is that, though the released dataset contains individual records, the accuracy of the relationships within a particular record can be eroded. Consider that the set of unidentified tables are registries of collected unidentified information, such as IP addresses that visit a particular website. One possible solution for privacy protection is to perturb the IP addresses as if they are categorical values (Evmimievski, Srikant, Agrawal, & Gehrke, 2002). Though perturbing an IP address ip into ip' can protect the identity of ip , it can falsely denote ip' as a visitor of the website. Furthermore, if ip' is listed in a released collection from another website, then a false trail may be established representing multiple addresses. Conversely, if the perturbation schema is not strong enough, then the creation of trails through probabilistic means may defeat the protection afforded by noise addition (discussed below). Similar problems may arise in the perturbation of names or identities.

A second set of potentially useful methods stems from computational disclosure control techniques (Domingo-Ferrer & Mateo-Sanz, 2002; Domingo-Ferrer & Torra, 2001; Hundepool & Willenborg, 1996; Sweeney, 2002a). The goal of such techniques is to prevent the direct linking through formal models of computability. One promising area of research to protect against trail re-identification is k -anonymity (Hundepool & Willenborg, 1996). For a dataset to be k -anonymous, each released record must be indistinguishable (not necessarily identical) as $k-1$ other records on a specified set of attributes. In previous methods, records were transformed through generalization and suppression on a hierarchy of values for a predefined set of attributes (Sweeney, 2002b). However, this raises several questions. First, what is the proper hierarchy of values? Consider two IP addresses $ip_1 = 128.2.52.132$ and $ip_2 = 128.2.53.151$. One possible generalization is the value "128.2.52.132 OR 128.2.53.151", with an alternative generalization simply 128.2.5x.xxx, where x is a wildcard representing any integer in the range 0 to 9. While the first seems more semantically correct, the second may be more amendable to known data mining tools. A second question is to what extent does k -anonymity affect the ability to learn useful patterns in the aggregate data? This problem is similar in nature to the problem of perturbation and correct aggregate reconstruction. Furthermore, we must address the question of how well anonymity, as well as sensitive information, is protected in the face of pattern recognition and inference techniques (Ohno-Machado, Silveira, & Vinterbo, 2004; Vinterbo, 2004).

A third area of potentially useful research is in secure multiparty computation. The goal of such procedures is to learn novel information from multiple parties' datasets while revealing minimal information about one party's dataset to another party. The work most related to the trail problem is that of the learning algorithms for horizontal- (same attributes - different transactions) (Kantarcioglu & Clifton, 2004; Kantarcioglu & Vaidya, 2004; Malin, Airodi, Edoho-Eket, & Li 2005), and vertical-partitioned (same transactions - different attributes) (Lindell & Pinkas, 2001; Vaidya & Clifton, 2002). Rather than release

plaintext data directly, cryptographic approaches are used to learn aggregate association rules or construct classifiers such as decision trees. While privacy may be preserved using these techniques, neglecting leakage via collusion and other features, no specific information is ever released from an institution. Yet, in the trail problem, data may need to be revealed by the websites to outside groups, especially when public use files need to be made available. Regardless, it may be that variants of multiparty computation can be utilized to determine what data should and should not be released from each particular location.

6.2 Extending Trail Re-identification

The above discussion on related re-identification research provides a partial roadmap for extending this work. As such, this section touches upon several brief suggestions.

The REIDIT algorithms are deterministic in nature due to assumptions made over the truthfulness of the data and the manner by which data is collected. One of the core simplifying assumptions is in the use of trails of binary strings without any error. An obvious extension of our research is in the design and evaluation of models that allow for the probabilistic qualification of trail bits. Additionally, in future research we must address the case where neither track is reserved to other. For instance, in this case, one location may undercollect names on a certain portion of a population, but IP addresses on a different portion. This characterization of a trail opens up two branches of research. The first is to consider an adaptation of record linkage or pattern matching for trail re-identification. The second is to phrase the trail re-identification problem as an optimization problem for re-identification. Locations, or combinations of such, could be afforded more weighting than others. Moreover, given that protection schemas may employ the incorporation of noise, one could imagine the design and analysis of trail re-identification techniques with noise filtering.

It may be that a combination of these methods will be necessary to ensure the maximum utility of the data. In order to do so, such techniques may be dependent on the data type considered and continued research will be able to answer this question for certain. Regardless, it is apparent that formal analysis of how to share data while maintaining privacy in distributed data is necessary.

ACKNOWLEDGEMENTS

This paper is a culmination and extension of prior work written by the author and is the first publication of such. The initial research on trail re-identification by the author examined a simple notion of REIDIT-C to re-identify DNA sequences to their subjects (Malin & Sweeney, 2001). Brad Malin's Master's Thesis (Malin, 2002), an unpublished work, provided more rigorous analysis and presented the first version of the REIDIT-C, REIDIT-I and REIDIT-M algorithms. These algorithms were then applied narrowly to the re-identification of DNA sequences (Malin & Sweeney, 2002, 2004). In an unpublished work, these algorithms were naively applied to weblogs in order to re-identify on-line consumers to the IP addresses left behind at visited websites (Malin, Sweeney, Newton, 2003). This paper builds on these prior works by clarifying terms and assumptions, further developing the overarching framework in which these algorithms operate, and providing more rigorous analysis of their operation on real-world data. The author would like to thank Latanya Sweeney for numerous conversations on privacy, formal modeling, and algorithm analysis. Additional thanks are extended to the current and former members of the Data Privacy Laboratory for useful discussions. Additional thanks are extended to the editors and anonymous referees for their insightful comments and questions which helped to greatly increase both the clarity and quality of this paper. Gratitude is expressed toward Alan Montgomery, Robert Kraut, and the Homenet project for the use of their data. This work was funded by the Data Privacy Laboratory at Carnegie Mellon University.

References

Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 439-450, Dallas, Texas, 2000.

Mikhail Bilenko and Raymond Mooney. On evaluation and training-set construction for duplicate detection. In *Proceedings of the ACM SIGKDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation*. Washington, DC, 2003.

Lee Breslau, Pei Cao, Li Fan, Graham Phillips and Scott Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of IEEE INFOCOM*, pages 126-134, New York, New York, 1999.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual (web) search engine. *Computer Networks and ISDN Systems*, 30: 107-117, 1998.

The Bandwidth Report. May Bandwidth Report – US Broadband Penetration Breaks 35%. Accessed June 11, 2003. <http://www.websiteoptimization.com/bw/>.

Sumit Chowdhury, George Duncan, Ramayaa Krishnan, Steve Roehrig, and Sumita Mukherjee. Disclosure detection in multivariate categorical databases: auditing confidentiality protection through two new matrix operators. *Management Science*, 45(12): 1710-1723, 1999.

Lawrence Cox. Suppression methodology and statistical disclosure control. *Journal of the American Statistical Association*, 1980; 75: 377-385.

AG De Waal and Leon Willenborg. A view on statistical disclosure control for microdata. *Survey Methodology*, 22: 95-103, 1996.

Josep Domingo-Ferrer and Josep Mateo-Sans. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14(1): 189-201, 2002.

Josep Domingo-Ferrer and Vicenc Torra. Disclosure methods and information loss for microdata. In P. Doyle, J. Lane, J. Theeuwes, L. Zayatz (eds), *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*. North-Holland, pages 91-110, 2001.

George Duncan and Stephen Fienberg. A Markov perturbation method for tabular data, turning administrative systems into information systems. *IRS Methodology Report Series*, 5: 223-231, 1997.

Mohamed Elfeky, Vassilios Verykios, and Ahmed Elmagarmid. TAILOR: a record linkage toolbox. In *Proceedings of the 18th International Conference on Data Engineering*, pages 17-28, San Jose, California, 2002.

Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. Privacy preserving mining of association rules. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217-228, Edmonton, Canada, 2002.

Ivan Fellegi and Alan Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64: 1183-1210, 1969.

Anco Hundepool and Leon Willenborg. μ - and τ -argus: software for statistical disclosure control. In *Proceedings of the Third International Seminar on Statistical Confidentiality*, Bled, Slovenia, 1996.

Industry Analysis and Technology Division, Wireline Competition Bureau. High Speed Services for Internet Access: Status as of June 20, 2002. Federal Communications Commission. Dec 2002.

Murat Kantarcioglu and Chris Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering*, 16(9): 1026-1037, 2004.

Murat Kantarcioglu and Jaideep Vaidya. Privacy preserving naïve bayes classifier for horizontally partitioned data. In *Proceedings of the 2004 IEEE ICDM Workshop on Privacy, Security, and Data Mining*. Melbourne, Florida, 2004.

Robert Kraut, Sara Kiesler, Bonka Boneva, Jonathon Cummings, Vicki Helgeson, and Anne Crawford. Internet paradox revisited. *Journal of Social Issues*, 58: 49-74, 2002.

- Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3): 177-206, 2002.
- Bradley Malin. Compromising privacy with trail re-identification: the REIDIT algorithms. Technical Report CMU-CALD-02-108, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2002.
- Bradley Malin, Edoardo Airoldi, Samuel Edoho-Eket, and Yiheng Li. Configurable security protocols for multi-party data analysis with malicious participants. In *Proceedings of the 21st IEEE International Conference on Data Engineering*, Tokyo, Japan, 2005; 533-544.
- Bradley Malin and Latanya Sweeney. Re-identification of DNA through an automated process. In *Proceedings of the American Medical Informatics Association Annual Symposium*, pages 423-427, Washington, DC, 2001.
- Bradley Malin and Latanya Sweeney. Compromising privacy in distributed population-based databases with trail matching: a dna example. Technical Report CMU-CS-02-189, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2002.
- Bradley Malin and Latanya Sweeney. How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. *Journal of Biomedical Informatics*, 37(3): 179-192, 2004.
- Bradley Malin, Latanya Sweeney, and Elaine Newton. Trail re-identification: learning who you are from where you have been. Technical Report LIDAP-WP12, Data Privacy Laboratory, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2003.
- Howard Newcombe, James Kennedy, S Axford, and A James. Automatic linkage of vital records. *Science*, 130: 954-959, 1959.
- Lucila Ohno-Machado, Paulo Silveria, and Staal Vinterbo. Protecting patient privacy by quantifiable controls of disseminated databases. *International Journal of Medical Informatics*, 73(7-8): 599-606, 2004.
- Paul Rappoport, Donald Kridel, and Lester Taylor. The demand for broadband: access, content, and the value of time. In Robert Crandal, ed. *Broadband: Should We Regulate High-Speed Internet Access?* Brookings Institution Press. Washington, DC, 2002.
- Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269-278, Edmonton, Canada. 2002.
- Latanya Sweeney. Uniqueness of simple demographics in the US population. Technical Report LIDAP-WP4, Data Privacy Laboratory, Carnegie Mellon University, Pittsburgh, Pennsylvania. 2000.
- Latanya Sweeney. Information explosion. In L. Zayatz, P. Doyle, J. Theeuwes and J. Lane (eds): *Confidentiality, disclosure, and data access: theory and practical applications for statistical agencies*, pages 43-74, Urban Institute, Washington, DC, 2001.
- Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(7): 557-570, 2002.
- Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(7): 571-588, 2002.
- Vicenc Torra. Re-identifying individuals using OWA operators. In *Proceedings of the Sixth International Conference on Soft Computing*. Iizuka, Japan. 2000.
- Vicenc Torra. Towards the re-identification of individuals in data files with non-common variables. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 326-330, Berlin, Germany, 2000.
- Vicenc Torra. OWA operators in data modeling and re-identification. *IEEE Transactions on Fuzzy Systems*, 12(5): 656-660, 2004.

Jaideep Vaidya and Chris Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 639-644, Edmonton, Canada, 2002.

Staal Vinterbo. Privacy: a machine learning approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(8): 939-948, 2004.

William Winkler. Matching and record linkage. In Brenda Cox, editor: *Business Survey Methods*, pages 355-384, J. Wiley, New York, 1995.

William Winkler. Re-identification methods for evaluating the confidentiality of analytically valid microdata. In J. Domingo-Ferrer, editor: *Statistical Data Protection*. Luxemburg, 1999.

William Winkler. Methods for record linkage and Bayesian networks. Technical Report Statistics-2002-05, Statistical Research Division, U.S. Census Bureau, Washington, DC. 2002.