B. Malin, L. Sweeney, and E. Newton. *Trail re-identification: learning who you are from where you have been*. LIDAP-WP12. Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA: March 2003.

Trail Re-Identification: Learning Who You Are From Where You Have Been

Bradley Malin
Data Privacy Laboratory
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
malin@cs.cmu.edu

Latanya Sweeney
Data Privacy Laboratory
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
latanya@privacy.cs.cmu.edu

Elaine Newton
Data Privacy Laboratory
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
enewton@privacy.cs.cmu.edu

ABSTRACT

This paper provides algorithms for learning the identities of individuals from the trails of seemingly anonymous information they leave behind. Consider online consumers, who have the IP addresses of their computers logged at each website visited. Many falsely believe they cannot be identified. The term "re-identification" refers to correctly relating seemingly anonymous data to explicitly identifying information (such as the name or address) of the person who is the subject of those data. Re-identification has historically been associated with data released from a single data holder. This paper extends the concept to "trail re-identification" in which a person is related to a trail of seemingly anonymous and homogenous data left across different locations. The 3 novel algorithms presented in this paper perform trail re-identifications by exploiting the fact that some locations also capture explicitly identifying information and subsequently provide the unidentified data and the identified data as separate data releases. Intersecting occurrences in these two kinds of data can reveal identities. For example, an online consumer may visit 50 websites and purchase at 5 and another may visit 30 sites and purchase at 7. Shared visit logs provide unidentified data. Exchanged customer lists provide identified data. The algorithms presented herein re-identify individuals based on the uniqueness of trails across unidentified and identified datasets. The algorithms differ in the amount of completeness and multiplicity assumed in the data. Successful re-identifications are reported for DNA sequences left by hospital patients and for IP addresses left by online consumers. These algorithms are extensible to tracking collocations of people, which is an objective of homeland defense surveillance.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems – *Distributed databases;* H.2.8 [**Database Management**]: Database Applications – *Data mining*.

General Terms

Keywords

Re-identification Algorithms, Distributed Databases, Homeland Defense, Security and Privacy

1. INTRODUCTION

As a person progresses through daily life, they leave behind fragments of information about themselves in all kinds of disparate databases [9]. Examples include images of the same car recorded on different highway video cameras; a computer's IP address logged at multiple websites; and, a patient's DNA sequence appearing in different hospital databases. Much like fingerprint evidence in an earlier time, tiny pieces of digital information that are left behind seem innocent and anonymous. No one expects to easily relate these pieces of data to identities of people. Perhaps automated re-identification did not seem realistic before, but this work provides real-time algorithms for learning identities of people from the patterns of data pieces left across multiple locations.

Until recently, it was believed that if data looked anonymous, it was anonymous. Tables, in which each row of information related to a person, were shared somewhat freely provided none of the columns included explicit identifiers, such as name, address, or Social Security number. This kind of "de-identified" data can often be linked to other tables that do include explicit identifiers ("identified data") 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 [10].

As just described, "re-identification" was limited to relating people to records in a single released table, where the table was information collected by one data holder. In this work, the notion of re-identification is extended to "trail re-identification" which seeks to identify people who visited named locations. In trail re-identification, each location separately collects and subsequently shares de-identified data on people who visited the location. The de-identified data consists of only one or very few fields. A location may also share explicitly identified data, thereby naming some people. Recognizing uniquely occurring visit patterns across both the de-identified and identified datasets provides the basis for trail re-identification.

For example, an online consumer visits websites, leaving the IP address of his computer logged at each site visited. At some sites, he may also provide explicitly identifying information; for example, his name and address are provided to complete a purchase. Separately, these websites may share logs containing the IP addresses of those who visited their sites. As businesses, these websites may also share explicitly identified data such as customer lists, which typically includes the name and address 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. Until this work, many people believed these de-identified logs of IP addresses could not be re-identified.

As a second example, a patient sick with a deterministic genetic disorder may visit several hospitals. Each hospital records demographic information, clinical data and a digitally recording of the patient's DNA. Later, researchers share de-identified data consisting of only the DNA sequences collected at hospitals [1]. Many states sell hospital discharge data, which includes identifiable demographics and clinical information on each patient [9]. By examining the trails of which DNA sequences appeared at which hospitals in the de-identified DNA data and matching those visit patterns to which patients appeared at which hospitals in the identifiable hospital discharge data, DNA sequences can relate to names and addresses. Until this work, many people falsely believed that a DNA sequence could not be re-identified to a person in the absence of a master DNA registry.

There is ever-increasing demand to learn more about individuals from data people leave behind. For example, in homeland defense surveillance, learning which people have appeared at the same locations during similar time periods is very important. These kinds of trail re-identification problems represent an important extension to KDD research. This paper introduces the general trail re-identification problem and several of its variants. Three novel algorithms are provided. Results are reported on real-world datasets. This paper ends with a discussion on the implications of the trail re-identification problem to privacy and surveillance and on directions for future work.

2. DATA AND TRAIL DEFINTIONS

This section defines terms. Definitions begin with de-identified and identified tables. Different ways data can be released and the different kinds of trails that result are carefully described. The section ends with a formal definition of the trail re-identification problem. In the next section then, 3 algorithms that solve variants of the trail re-identification problem are presented.

The basics elements are derived from relational database theory. The term "data" refers to information held by a data-collecting location. The information is organized as a table of rows (records) and columns (fields). Each row (or "tuple") is specific to a person, machine, or other entity that visited the location. Each column is referred to as an attribute, which contains a information that refers to people, machines or other entities that visited the location. A table is defined as $\boldsymbol{\tau}(A_1, A_2,\ldots,A_p)$, where the set of attributes for table $\boldsymbol{\tau}$ is $A = \{A_1, A_2,\ldots,A_p\}$. A tuple t of the table $\boldsymbol{\tau}$ is defined as $t[a_1,\ldots,a_p]$ and represents the sequence of values, $v_1 \in A_1,\ldots,v_p \in A_p$. The size of the table is simply the number of tuples and is represented $|\boldsymbol{\tau}|$.

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 (de-identified). The properties of the partitioned release are formalized in definition 2.1. A sample is provided in Example 2.1.

Definition 2.1 (De-identified and Identified Tables). Given a table $\boldsymbol{\tau}(A_1, A_2,\ldots,A_p)$ maintained by a data-collecting location, the attributes $A^- = \{A_i,\ldots,A_j\} \subseteq A_1,\ldots,A_p$ and $A^+ = \{A_x,\ldots,A_y\} \subseteq A_1,\ldots,A_p$, $\boldsymbol{\tau}^-$ is the *de-identified* subtable of $\boldsymbol{\tau}$ having attributes A^-, and $\boldsymbol{\tau}^+$ is the *identified* subtable of $\boldsymbol{\tau}$ having attributes A^+ such that:

i) $A^- \cap A^+ = \varnothing$
ii) A^- is devoid of:
  a) explicitly identifying attributes; and,
  b) attributes linkable to an explicit identifier
iii) A^+ includes:
  a) explicitly identifying attributes; or,
  b) attributes linkable to an explicit identifier

The third part of Definition 2.1 states that the identified table is identifiable if it contains any explicitly identifying attributes, such as name or address, or if it has attributes that can be linked to any external tables which contain explicitly identifying attributes.

Example 2.1 Figure 1 illustrates released partitions from the hospital data in Figure 2. Patient demographics are reported in the identified table $\boldsymbol{\tau}^+$. DNA sequences are reported in the de-identified table, $\boldsymbol{\tau}^-$. The attribute ID is not released. The tuple for "Fran Booth" (shown in Figure 2) is not released.

$\boldsymbol{\tau}^+$			
Name	**Birthdate**	**Gender**	**Zip**
John Smith	2/18/45	M	15234
Mary Doe	4/9/75	F	15097
Bob Little	2/26/49	M	15212
Kate Erwin	11/3/54	F	15054

$\boldsymbol{\tau}^-$
DNA
acag…t
accg…a
cttg…a
atcg…t

Figure 1. Vertical partitioning by a hospital of its data into an identified table ($\boldsymbol{\tau}^+$) of patient demographics and a de-identified table ($\boldsymbol{\tau}^-$) containing DNA sequences.

As described in Definition 2.1, a vertical partitioning involves a pair of functions V_{id} and V_{de} such that V_{id}: $\boldsymbol{\tau} \rightarrow \boldsymbol{\tau}^+$ and V_{de}: $\boldsymbol{\tau} \rightarrow \boldsymbol{\tau}^-$. In compliance with the relational model, the order in which tuples appear in the de-identified and identified tables is not necessarily maintained across the tables.

Notice that definition 2.1 does not require all the tuples in $\boldsymbol{\tau}$ to be contained in the identified table ($\boldsymbol{\tau}^+$) or in the de-identified table ($\boldsymbol{\tau}^-$). Constraints on the relationships between the number and containment of tuples provided in released tables are discussed in the next section.

2.1 Ways of Releasing Data

Here are two ways a data-collecting location can specify tuples for partitioning. The location can include attributes from the same, and only the same, tuples in both the identified table and the de-identified table. Alternatively, the location can include attributes for a subset of tuples in the de-identified table to appear in the identified table, or vice versa. Precise descriptions of these two

kinds of releases are provided in this subsection, but first, assumptions inherent in this work are stated.

Assumption 2.1 (Per Location Release). Each data-collecting location *c* releases data that was collected at *c* and from no external source.

Assumption 2.2 (Uniqueness of Tuples). In a location's de-identified and identified tables, each tuple is unique. Therefore, a de-identified or identified table represents a set of references to people, machines, or other entities that have "visited" the location, but not necessarily the frequency of visits. These references narrowly relate to a person, machine, household or other entity to be identified.

Definition 2.2 presents a definition for released data that adheres to a "representative" property in which only tuples present in the de-identified table have corresponding tuples in the identified table, and vice versa. Example 2.2 provides a sample.

Definition 2.2 (Representative) Let the table $\boldsymbol{\tau}$ be vertically partitioned by V_{id} and V_{de} such that V_{id}: $\boldsymbol{\tau}\rightarrow\boldsymbol{\tau}^+$ and V_{de}: $\boldsymbol{\tau}\rightarrow\boldsymbol{\tau}^-$, where $\boldsymbol{\tau}^-$ is the de-identified table, and $\boldsymbol{\tau}^+$ is the identified table. The tables $\boldsymbol{\tau}^-$ and $\boldsymbol{\tau}^+$ are *representative* if and only if: (1) $\forall t_{id}\in\boldsymbol{\tau}^+$ and $\forall t_{de}\in\boldsymbol{\tau}^-$, $V_{id}^{-1}(t_{id}) = V_{de}^{-1}(t_{de})$; and, (2) $|\boldsymbol{\tau}^+| = |\boldsymbol{\tau}^-|$. [1]

In releases that adhere to the representative property, every tuple from the data-collecting location present in the de-identified table is also present in the identified table, and vice versa.

Example 2.2 Figure 1 depicts a representative release resulting from a representative vertical partitioning of the data in Figure 2 into a de-identified table ($\boldsymbol{\tau}^-$) and an identified table ($\boldsymbol{\tau}^+$). Each released tuple has values in both tables.

Name	Birthdate	Gender	ID	Zip	DNA
John Smith	2/18/45	M	11	15234	acag…t
Mary Doe	4/9/75	F	18	15097	accg…a
Bob Little	2/26/49	M	2	15212	cttg…a
Kate Erwin	11/3/54	F	21	15054	atcg…t
Fran Booth	1/8/71	F	27	15054	accg…t

Figure 2. Original data collected by a hospital.

Releases that are representative are not always practical. In some situations, a data-collecting location may not have collected both identified and de-identified data on all visitors to their location 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 "appropriate" (Definition 2.3). Samples are provided in Examples 2.3 and 2.4.

Definition 2.3 (Appropriate) Let the table $\boldsymbol{\tau}$ be vertically partitioned by V_{id} and V_{de} such that V_{id}: $\boldsymbol{\tau}\rightarrow\boldsymbol{\tau}^+$ and V_{de}: $\boldsymbol{\tau}\rightarrow\boldsymbol{\tau}^-$, where $\boldsymbol{\tau}^-$ is the de-identified table, and $\boldsymbol{\tau}^+$ is the identified table. The tables $\boldsymbol{\tau}^-$ and $\boldsymbol{\tau}^+$ are *appropriate* if either (1) $\forall t_{id}\in\boldsymbol{\tau}^+$, there exist $t_{de}\in\boldsymbol{\tau}^-$ such that $V_{id}^{-1}(t_{id}) = V_{de}^{-1}(t_{de})$; or, (2) $\forall t_{de}\in\boldsymbol{\tau}^-$, there exist $t_{id}\in\boldsymbol{\tau}^+$ such that $V_{id}^{-1}(t_{id}) = V_{de}^{-1}(t_{de})$. In (1), $\boldsymbol{\tau}^+$ is the "appropriate" table to $\boldsymbol{\tau}^-$; and in (2), $\boldsymbol{\tau}^-$ is the "appropriate" table to $\boldsymbol{\tau}^+$.

Example 2.3 Consider an online store in which all purchases are made at the store's website. An online consumer may visit the store and not necessarily make a purchase; of so, his de-identified IP address may be collected, but there is no accompanying name and address due to lack of purchase. The store's release of all names of purchasers as the identified table and all logged IP addresses as the de-identified table is a release that adheres to the appropriate property.

Example 2.4 Figure 3 depicts a release resulting from an appropriate vertical partitioning of the data in Figure 2 into a de-identified table ($\boldsymbol{\tau}^-$) and an identified table ($\boldsymbol{\tau}^+$). DNA sequences for "Bob Little" and "Kate Erwin" appear in $\boldsymbol{\tau}^-$, but there are no DNA sequences for "John Smith" or "Mary Doe."

$\boldsymbol{\tau}^+$			
Name	**Birthdate**	**Gender**	**Zip**
John Smith	2/18/45	M	15234
Mary Doe	4/9/75	F	15097
Bob Little	2/26/49	M	15212
Kate Erwin	11/3/54	F	15054

$\boldsymbol{\tau}^-$
DNA
cttg…a
atcg…t

Figure 3. Release by a hospital, with partitioning into an identified table ($\boldsymbol{\tau}^+$) of patient demographics and an appropriate de-identified table ($\boldsymbol{\tau}^-$) containing DNA sequences.

2.2 Data Trails

Given a set of data-collecting locations, where all locations share either releases in a representative or an appropriate 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 a de-identified track and an identified track, respectively, (Definition 2.4).

Definition 2.4 (De-identified and Identified Tracks) Let C be the set of data-collecting locations that share their identified tables, $\boldsymbol{\tau}_c^+$, over the attributes A^+ and de-identified tables, $\boldsymbol{\tau}_c^-$, over the attributes A^-, where $c\in C$. Let B be a vector containing the members of C, and T^+ be the set of all $\{\boldsymbol{\tau}_c^+\}$ and T^- be the set of all $\{\boldsymbol{\tau}_c^-\}$ for each $c\in C$. Either: (1) $\boldsymbol{\tau}_c^+$ and $\boldsymbol{\tau}_c^-$ are representative; (2) $\boldsymbol{\tau}_c^+$ is appropriate to $\boldsymbol{\tau}_c^-$ for each $c\in C$; or, (3) $\boldsymbol{\tau}_c^-$ is appropriate to $\boldsymbol{\tau}_c^+$ for each $c\in C$. The *de-identified track*, **N,** is a matrix having $|A^-| + |C|$ 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 $|A^+| + |C|$ columns and the contents of **P** are the same as those realized by *FillTrack*(**P**, A^+, T^+). The number of rows in **N** and **P** are:

$$\left|\bigcup_{c=1}^{|C|}\tau_c^-\right| \text{ and } \left|\bigcup_{c=1}^{|C|}\tau_c^+\right|, \text{ respectively.}$$

FillTrack(Track T, Attributes A, Tables $\{\boldsymbol{\tau}_c\}$)

Steps:
- **let** each cell in **T** be initialized to 0
- **for** each location $c\in C$:
  - **For** each tuple $t_i\in\boldsymbol{\tau}_c$
    - **let** b be the index of c in **B**
    - **if** there does not exist $\mathbf{T}_j[1,\ldots,|A|] \equiv t_i$, where j=1,…,|**T**|
    - **then**:
      - **let** k be the first unused row in **N** *// has all 0's*
      - $\mathbf{N_k}[1,\ldots,|A|] = t_i$ and $\mathbf{N_k}[|A|+b] = 1$
    - **Else**: $\mathbf{N_j}[|A|+b] = 1$ *// another location found*

*// $\boldsymbol{N_x}[y]$ is the cell in the x-th row and y-th column of **N***
*// $\boldsymbol{N_x}[a,...,b]$ is the vector in the x-th row having columns a to b of **N***

[1] V^{-1} is the inverse function of V.

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 locations are 1 if the subject visited the location and a 0 otherwise.

In a representative release, the identified track (**P**) and the de-identified track (**N**) are "representative." If the tables that construct **N** are each appropriate to the tables that constitute **P**, track **N** is "appropriate" to **P**. Likewise, if the tables that construct **P** are each appropriate to the tables that constitute **N**, track **P** is "appropriate" to **N**. Examples 2.5 and 2.6 provide samples.

Example 2.5 Figure 5 shows the identified track (**P**) and de-identified track (**N**) for the releases found in Figure 4, which adhere to the representative property.

P			
Name	h_1	h_2	H_3
John	1	1	0
Mary	1	0	1
Bob	0	1	1
Kate	0	0	1

N			
DNA	h_1	h_2	h_3
acag…t	1	1	0
accg…a	1	0	1
cttg…a	0	1	1
atcg…t	0	0	1

Figure 4. De-identified track (N) and identified track (P).

Example 2.6 Figure 6 shows 3 hospitals performing appropriate releases in which the de-identified DNA tables are appropriate to the tables of names. These releases provide the tracks in Figure 7 in which track **N**, which results from the DNA tables, is appropriate to **P**, which results from the tables of names.

In both a de-identified track and an identified track, the rightmost columns are associated with locations. The vectors of binary values associated with those columns are "trails." They show the locations where the person, machine, or entity that is the subject of the visits has been. Trails are described in Definition 2.5.

Definition 2.5 (Trail) Let C be the set of data-collecting 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 $\mathbf{T}_j[|A|+1,\ldots,|A|+|C|]$. For convenience, $\mathbf{T}_j[|A|+1,\ldots,|A|+|C|]$. is written *trail*(**T**, j).

A trail for a subject is a vector of binary values where a value of 1 indicates the subject visited the location and 0 otherwise. See Example 2.7.

Hosp1 (+)
Name
John
Mary

Hosp1 (-)
DNA
acag…t
accg…a

Hosp2 (+)
Name
John
Bob

Hosp2 (-)
DNA
acag…t
cttg…a

Hosp3 (+)
Name
Mary
Bob
Kate

Hosp3 (-)
DNA
accg…a
cttg…a
atcg…t

Figure 5. Releases by 3 hospitals that adhere to the representative property.

Example 2.7 Given the identified track **P** in Figure 5, [1,1,0] is a trail for "John" and [0,1,1] is a trail for "Bob". Given the de-identified track **N** in Figure 5, [1,0,1] is a trail for "accg…a" and [0,0,1] is a trail for "atcg…t".

The de-identified and identified tracks in Figure 5 were constructed from the releases reported in Figure 4. These tracks adhere to the representative property, and the resulting trails are "complete trails". The notion of a complete trail is presented in Definition 2.6.

Definition 2.6 (Complete Trail) Let C be the set of data-collecting locations that share their identified (or de-identified) tables in track **T** such that the shared tables from each location $c \in C$ is representative. 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.

If de-identified and identified tracks are constructed from a release that adheres to the appropriate property, then the trails in the appropriate track are all "incomplete trails." Definition 2.7 presents an incomplete trail and samples are provided in Example 2.8.

Definition 2.7 (Incomplete Trail) Let C be the set of data-collecting locations that share their identified and de-identified tables in tracks $\mathbf{T_1}$ and $\mathbf{T_2}$ such that $\mathbf{T_1}$ is the appropriate track of $\mathbf{T_2}$ for all data holdrs $c \in C$. An *incomplete trail* is a trail in $\mathbf{T_1}$. 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.

Hosp1 (+)
Name
John
Mary

Hosp1 (-)
DNA
acag…t

Hosp2 (+)
Name
John
Bob

Hosp2 (-)
DNA
acag…t
cttg…a

Hosp3 (+)
Name
Mary
Bob
Kate

Hosp3 (-)
DNA
atcg…t

Figure 6. Released identified tables (+) and de-identified tables (-) with the appropriate property.

Example 2.8 In Figure 7 the de-identified track **N** has the following incomplete trails: [0,1,0]; [1,0,0]; [0,1,0]; and [0,0,1]. All the trails in the identified track **P** are complete.

P			
Name	h_1	h_2	h_3
John	1	1	0
Mary	1	0	1
Bob	0	1	1
Kate	0	0	1

N			
DNA	h_1	h_2	h_3
acag…t	0	1	0
accg…a	1	0	0
cttg…a	0	1	0
atcg…t	0	0	1

Figure 7. De-identified track (N) and identified track (P) from the partitions in Figure 6. P has complete trails. N has incomplete 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 2.9.

Definition 2.8 (Subtrails / Supertrails) Let C be the set of data-collecting locations that share their identified and de-identified

tables in tracks $\mathbf{T_1}$ and $\mathbf{T_2}$ such that $\mathbf{T_1}$ is appropriate to $\mathbf{T_2}$. The shared tables are over the attributes A. Let x be a trails from $\mathbf{T_1}$ and y be a trail from $\mathbf{T_2}$. x is a *subtrail* of y (written $x \le y$)and y is a *supertrail* of x (written $y \ge x$) if and only if: $\mathbf{T_1}[x]\ [d] \le \mathbf{T_2}\ [y][d]$ for $d=|A|+1,\ldots,|A|+|C|$.

Example 2.9 [1,0,0], [0,1,0], and [1,1,0] are subtrails of [1,1,0]. [1,1,0] and [0,1,1] are supertrails of [0,1,0].

With respect to tracks, recall the properties of representative (2.2) and appropriate (2.3). If tracks **N** and **P** are constructed from representative releases, then for any particular entity x in the tracks, $\mathbf{N}[x][d]$ must equal $\mathbf{P}[x][d]$ for all locations d. Furthermore, if **N** and **P** are constructed from releases that are appropriate, then for any particular entity x in the tracks, $\mathbf{N}[x][d]$ must be $\le \mathbf{P}\ [x][d]$ for all locations d if **N** is appropriate to **P.** **N** consists of incomplete trails and **P** consists of complete trails. The converse is true if **P** is appropriate to **N.**

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. 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**.

Now that de-identified and identified tracks are understood and how complete and incomplete trails relate to these tracks, the trail re-identification problem can be presented. See Definition 2.9 and Example 2.10.

Definition 2.9 (Trail Re-identification Problem) Let C be 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. Let there exist a function f: $\mathbf{A}\rightarrow\mathbf{B}$, where $\mathbf{A}\in\{\mathbf{N}, \mathbf{P}\}$ and $\mathbf{B}=\{\mathbf{N}, \mathbf{P}\}-\{\mathbf{A}\}$. A trail re-identification results for a subject s when there exists an i, such that $f(\mathbf{A_s}[1,\ldots,|A^-|]) = \mathbf{B_i}[1,\ldots,|A^+|]$. The goal is to determine the proper function f.

Example 2.10 The de-identified track **N** and identified track **P** in Figure 4 result from the hospital releases shown in Figure 5. f(["acag…t"]) = ["John"], f(["accg…a"]) = ["Mary"], f(["cttg…a"]) = "Bob"], and f(["atcg…t"]) = ["Kate"] are all correct trail re-identifications.

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. Afterwards, results are reported on real-world data. The paper ends with discussions on related work and pertinent issues.

3. REIDIT ALGORITHMS

Given C, 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. It performs exact match on the trails of **N** and **P**. REIDIT-C assumes that both **N** and **P** are representative, and therefore, REIDIT-C only works on complete trails.

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 8.

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 $O(|\mathbf{N}|\bullet|\mathbf{P}|)$ or $O(|\mathbf{N}|^2)$ because $|\mathbf{N}|=|\mathbf{P}|$. This is an artifact of the way in which the pseudo code is written. Another version could be written in which each set of trails are sorted and then compared, resulting in $O(|\mathbf{N}|\log|\mathbf{N}|)$.

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 trail re-identifications R

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

```
Steps:
    let R = ∅
    for n=1 to |N|
        let M = ∅
        for p= 1 to |P|
            if trail(N,n) ≡ trail(P,p)
                M = M ∪ P_p[1,…,|A+|]
                s = p
        if |M| ≡ 1
            R = R ∪ {(P_s[1,…,|A+|], N_n[1,…,|A-|])}
    return R
```

Figure 8. Pseudocode for REIDIT-C.

Theorem 3.1 Trail re-identifications from REIDIT-C are correctly re-identified.

PROOF: First, recall the underlying assumption of the complete-release model: tuples of both tables **N** and **P** consist only of complete trails. Thus, at location i, a visit from an entity must be recorded in both $\mathbf{T_i^-}$ and $\mathbf{T_i^+}$. 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. ■

3.2 REIDIT-Incomplete

The second algorithm is named REIDIT-I. It performs subtrail/suptertrail matching on the trails of **N** and **P**. REIDIT-I assumes either **N** is appropriate to **P** or **P** is appropriate to **N**.

When an incomplete trail can be matched to a single complete trail, a trail re-identification occurs. Unlike REIDIT-C, equality cannot be used 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 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** have no more trails to process; or, (2) there are no re-identifications made in the current iteration. REIDIT-I appears in Figure 9.

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. Finally, the nested for process continues until no re-identifications are made during the while loop and the while may iterate a maximum of |**N**|. times. This provides $O(|\mathbf{N}|^2 \bullet |\mathbf{P}|)$.

Algorithm: REIDIT-I (X, Y)

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 appropriate table of **N** or **P** and **Y** is the other table.
Output: Set of trail re-identifications *R*
Assumes: 1) **X** has incomplete trails and **Y** has complete trails. 2) **X** and **Y** are one-to-one.

```
Steps
    let R = ∅
    Do
        FoundOne = False
        for n=1 to |X|
            let M = ∅
            for p= 1 to |Y|
                if trail(X,n) ≠ [null, …, null]
                  and trail(Y,p) ≠ [null, …, null]
                  and  trail(X,n) ≤ trail(Y,p)
                      M = M ∪ Y_p[1,…,|A+|]
                      s = p
            If |M| ≡ 1
                R = R ∪ {(Y_s[1,…,|A+|], X_n[1,…,|A-|])}
                X_n[|A-|+1,…,|A-|+|C|] = [null, …, null]   // remove
                Y_s[|A+|+1,…,|A+|+|C|] = [null, …, null]   // remove
                FoundOne = True
    while X has non-null tuples and FoundOne ≡ false
    return R
```

Figure 9. Pseudocode for REIDIT-I.

Theorem 3.2 Trail re-identifications from REIDIT-I are correctly re-identified.

PROOF: For convenience, assume that **N** is appropriate to **P**. **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 *M* (|*M*|≥1) for *trail*(**N**,*n*). If |*M*| 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 *M*. In the event when |*M*| > 1, then the algorithm can still converge to a correct re-identification as follows. Let |*M*| equal *k*. When a re-identification is made for a trail other than *trail*(**N**,*n*), then |*M*| decreases by 1. Because it is already known that |*M*| has a minimum of 1, if |*M*|-1 re-identifications are made for trails of **N**, excluding *trail*(**N**,*n*), each with a member from *M*, then the remaining member of *M* must re-identify *trail*(**N**,*n*). ■

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 appropriate to **P** or **P** is appropriate 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 10.

Algorithm: REIDIT-M (X, Y)

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 appropriate table of **N** or **P** and **Y** is the other table.
Output: Set of trail re-identifications *R*
Assumes: 1) **X** has incomplete trails and **Y** has complete trails. 2) **X** to **Y** is one-to-many.

```
Steps
    let R = ∅
    for n=1 to |X|
        let M = ∅
        for p= 1 to |Y|
            if trails(X,n) ≤ trails(Y,p)
                M = M ∪ Y_p[1,…,|A+|]
                s = p
        if |M| ≡ 1
            R = R ∪ {(Y_s[1,…,|A+|], X_n[1,…,|A-|])}
    return R
```

Figure 10. Pseudocode for REIDIT-M.

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 $O(|\mathbf{N}| \bullet |\mathbf{P}|)$.

4. THEORETICAL VS. ACTUAL RE-IDENTIFICATION

Theoretically, 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 *C*, if |**P**| ≤ |*C*|, then the maximum number of trail re-identifications is bounded by the number of subjects |**P**|, which implicates that all trails may be re-identified. When |**P**| > |*C*|, the maximum number of trail re-identifications is bounded by the number of locations in the exponential manner $2^{|C|}$-1. When $|\mathbf{P}| > 2^{|C|}$, 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**|.

There is evidence that a probabilistic model, such as a multinomial function over each data-collecting location, can be used to estimate the likelihood of a particular trail re-identification [10]. While an exponential number of trails may be constructed, only a fraction of the trails are ever observed. The probability of observing a trail is dependent on the number of subjects at each location. This theory is supported by empirical evidence from trail re-identifications of DNA sequence trails from patients with particular genetic disorders [4, 5].

5. RE-IDENTIFICATION EXPERIMENTS

To evaluate the REIDIT algorithms, real-world data sets corresponding to medical DNA database records and IP address data were used. Results are reported for each REIDIT algorithm in this section. The experiments are: REIDIT-C on DNA trails from patients, REIDIT-C, REIDIT-I, and REIDIT-M on online consumers, and an examination of REIDIT algorithms on very large datasets using synthetic data.

5.1 Complete DNA Re-identification

Description of Dataset. The dataset used is publicly available hospital discharge data from the State of Illinois, covering the years 1990 through 1997, with approximately 1.3 million hospital discharges per year [7]. Patient demographics, hospital identity, and diagnosis codes (including certain gene specific diseases) are among the attributes stored with each database entry. We make the assumption that hospitals collect the following two types of data: general hospital discharge information, which makes up the identified track **P** and, when possible, DNA sequences which constitute the basis for the de-identified track **N**.

There are more than 30 diseases with single gene determinants (i.e. if a person's gene is so mutated, the person deterministically gets the disease). For this study, we selected a subset of these to analyze, including Cystic Fibrosis, Friedreich's Ataxia, HeREIDITary Hemmorhagic Telangiectasia, Huntington's Disease, Phenylketonuria, Recsum's Syndrom, Sickle Cell Anemia, and Tuberous Sclerosis. In earlier work, the construction of clinical profiles from the hospital discharge data and inferences to DNA sequence samples was done [4, 5].

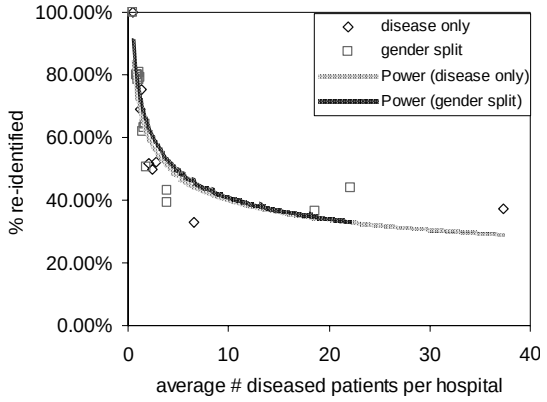

Figure 10. REIDIT-C Identity learning in DNA databases.

For this experiment, we assume that each hospital releases all patient discharge data, as mandated by state law, and each hospital releases all records from their DNA databases for research or clinical evaluation purposes. The attributes released with the de-identified tables were A^-={*DNA sequence*} and the attributes released with the identified tables were A^+={*date of birth, gender, zip code*} for each hospital location. In previous research [9, 10], the attributes A^+ were demonstrated to be re-identifiable. De-identified track **N** and identified track **P** were constructed specific to each of the 8 diseases. In all cases of **N** and **P**, they were representative. The number of rows in **N** (and **P**) ranged from 4 to 7730, depending on the disease. The number of hospital locations ranged from 8 to 207. REIDIT-C was used to perform trail re-identification on the DNA trails. Results are summarized in Figure 10, with and without the inference of gender from DNA sequences. We find that there is a power relationship (r^2=0.74 without gender, r^2=0.8 considering gender) between the average number patients per hospital and the percentage of the disease population that can re-identified.

5.2 Complete IP Re-identification

Description of Dataset. The second dataset was compiled by the Homenet project at Carnegie Mellon University[2] [3], who provide families in the Pittsburgh area with internet service in exchange for the monitoring and recoding of the families' online services and transactions. We used URL access data collected over a two-month period that included 86 households. We reconstructed purchase data and weblogs for websites accessed by this population. 5116 distinct websites and 66,862 distinct pages were accessed. 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.

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 $\mathbf{N_{per}}$ and identified track $\mathbf{P_{per}}$ 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. De-identified track $\mathbf{N_{hou}}$ and identified track $\mathbf{P_{hou}}$ were constructed having 26 rows. The number of locations was 24.

[2] For additional information about the Homenet project, we refer the reader to http://homenet.andrew.cmu.edu.

B. Malin, L. Sweeney, and E. Newton. *Trail re-identification: learning who you are from where you have been.* LIDAP-WP12. Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA: March 2003.

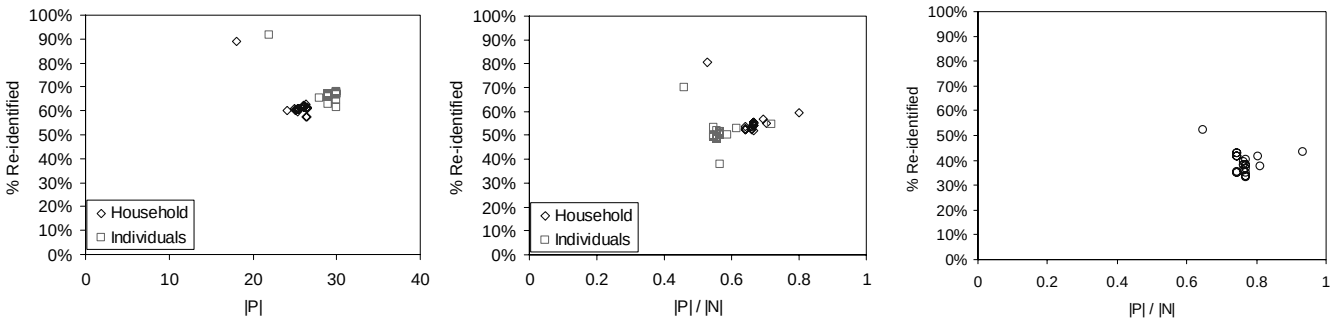

Figure 11. Sensitivity of re-identification to one location left out in data with left) a representative, middle) an appropriate property, and right) a multiple release. The x-axis in the left graph, and the y-axis in the remaining two, are slightly jittered for visual inspection.

REIDIT-C performed trail re-identifications on $\mathbf{N_{per}}$ and $\mathbf{P_{per}}$ and on $\mathbf{N_{hou}}$ and $\mathbf{P_{hou}}$; 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 (~62%) were re-identified to mailing address and 20 (~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 Figure 11.

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 no role in trail re-identification. Removal of this website allowed for ~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.

5.3 Incomplete IP Re-identification

Using the dataset described in section 5.2, 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}^{-}={*website*, *individual IP address*} and the attributes released with the identified tables remained A_{per}^{+}={*website*, *name*, *address*} for each targeted website location. De-identified track $\mathbf{N_{per}}$ had 53 rows and identified track $\mathbf{P_{per}}$ had 30 rows. $\mathbf{N_{per}}$ has incomplete trails. $\mathbf{P_{per}}$ 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}^{-}={*website*, *household IP address*} and the attributes released with the identified tables remained A_{hou}^{+}={*website*, *street address*} for each targeted website location. De-identified track $\mathbf{N_{hou}}$ had 39 rows and identified track $\mathbf{P_{hou}}$ had 26 rows. $\mathbf{N_{hou}}$ has incomplete trails. $\mathbf{P_{hou}}$ has complete trails. The number of locations remained 24.

Trail re-identification was done 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 (~35%) to households and 15 to individuals (~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 11. 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 ~25%.

5.4 Multiple IP Re-identification

Using the dataset described in section 5.2, 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^{-}={*website*, *IP address*} and the attributes released with the identified tables were A^{+}={*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 11.

5.5 Re-identification in Large Datasets

In this section we examine how the REIDIT algorithms scale to very large populations. To conduct these experiments we generated synthetic datasets with distributions based on those

found in the Homenet database; see section 5.2. Trails were simulated based on the probability that an individual visited a location. For complete trails the probability a trail position *i* equals 1 is (# visits at loc_i) / |**N**|. Incomplete trails were simulated from complete trails, by flipping 1 to 0 with probability equal to 1-[(# purchasers at loc_i) / (# visits at loc_i)]. We considered increases in the number of locations as a multiple *x* of the estimated probabilities, such that we concatenated *x* trails to consider a larger trail. Results for REIDIT-I are provided in Figure 12 for increasing size datasets. Holding visit and purchase probabilities constant, the algorithms scale to accommodate very large populations and numbers of locations. The number of trail re-identifications in a dataset decreases linear in a log scale of the size of the population. The slope decreases as the number of locations increases. For Figure 12, the slopes are approximately -0.27, -0.20, and -0.8, for 24, 48, and 72 purchasing locations, respectively, and continue to decrease with increasing numbers of locations. Similar linear scaling characteristics are found for REIDIT-C (not shown).

We have introduced a new kind of learning problem called trail re-identification. The identities of people, machines and other entities are found from fragments of information they leave across disparate locations. We introduced three novel algorithms for performing trail re-identifications based on finding uniqueness in visit patterns. While trail re-identification is new, other kinds of re-identification have been researched. The next section contains a survey of related work on re-identification.

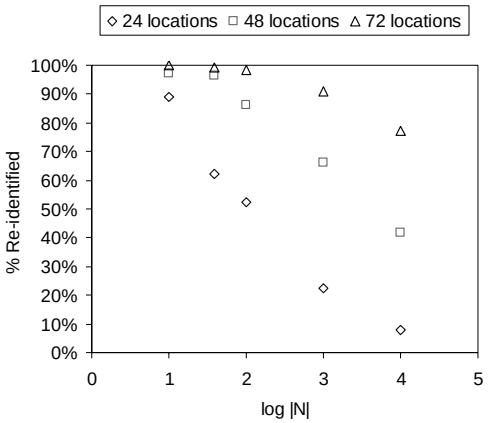

Figure 12. Scaling of REIDIT-I to changes in increased populations (in base 10 log scale) and number of locations.

6. RELATED RE-IDENTIFICATION RESEARCH

There have been several methods applied to the problem of re-identification. Mainly, the concept has been developed with respect to three genres: record linkage, data linkage, and pattern matching with aggregation operations. The techniques of record linkage were initially introduced by Newcombe [6], Felligi, and Sunter [2] and have been ushered into the modern statistical era by the work of Winkler [12]. The problem that record linkage 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×B → {*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. To do so, several assumptions are made about the data. First, it is assumed that there are two files with common variables and that there is typographical error in the files. Currently, record linkage methods employ expectation-maximization algorithms for converging to classifications of record pairs. 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.

Data linkage differs from record linkage in several fundamental aspects, most notably the fact that data linkage has been specifically designed for re-identification purposes. It is the intension of data linkage to make re-identifications for data devoid of an explicit identity. In addition, the attributes of the two files are not required to be the same, but instead it is concerned with exploiting inferential relations between attributes of the two files. A combination of the values in the attributes of a table is utilized to estimate the uniqueness of an entity's identity in a known population, beyond that of the considered files [8]. The addition of related attributes allows for an increased probability of the uniqueness of records, provided the added attributes can be related to features of the identified population. Linkage is established through known attributes. 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 of a large number of tables.

The third method of re-identification is based on ordered weighted aggregation (OWA) operators [11], which are rooted in the data mining community. While record and data linkage require that there exist direct inferential relationships between attributes of two tables, this approach attempts to re-identify when there are no common attributes. However, the technique requires several major assumptions. First, there is an assumption that there exist a large number of common individuals in the two datasets. Second, there exists an implicit similar structure to information in the two tables. Third, the datasets consist of numerical data. The procedure takes a table of records and attempts dimensionality reduction by converting the data vector V of a record [v_1, v_2, …, v_n] into a new vector W of several weighted scalars [w_1, w_2, …, 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 and the relationships of the individuals in the dataset on those combinations. Re-identification is then achieved by matching records 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. While the claim has been made that this technique can re-identify individual records in a table, corresponding to subjects and not attributes, no current research disputing or proving this claim exists.

7. CONCLUDING REMARKS 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 locations from small amounts of seemingly anonymous or innocuous evidence the person has been 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.

Only binary trails were examined in this work. However, one possible extension of this research is in the design and evaluation of models that allow for the probabilistic qualification of trail bits. This qualification would permit an interesting optimization problem for re-identification, allowing some locations to be weighted more than others. Future research will also need to address such issues as error and other kinds of incomplete and data quality issues.

The REIDIT algorithms are challenging and timely to society. The American public wants to feel safe and is therefore looking for protection through various kinds of electronic surveillance systems. An extension of the REIDIT algorithms can be used to track when suspicious people tend to travel together or be in the same places.

The REIDIT algorithms are also important to society because Americans are seeking more safety without compromising privacy unnecessarily. Clearly, the REIDIT algorithms exasperate 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. Currently, there is no work documenting how particular protection schemas might thwart the various trail re-identification methods presented herein. Because trail re-identification is a novel strategy for re-identification, it provides a new and important direction for future research in data privacy. 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. In this 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.

8. ACKNOWLEDGEMENTS

The authors would like to thank Yiheng Li, Robert Murphy, Rema Padman, and Victor Weedn for useful discussions. Additional thanks are extended to Alan Montgomery, Robert Kraut, and the Homenet project for the use of their data. We also thank Joe Lombardo and Julie Pavlin for their support. This work was funded by the Data Privacy Laboratory at Carnegie Mellon University.

REFERENCES

[1] Altman, R.B. and Klein, T.E. Challenges for biomedical informatics and pharmacogenomics. *Annu Rev Pharmacol Toxicol*, 42, 113–33, 2002.

[2] Fellegi, I.P., and Sunter, A.B. A theory for record linkage. *Journal of the American Statistical Association*. 64: 1183-1210, 1969.

[3] Kraut, R., Kiesler, S., Boneva, B., Cummings, J., Helgeson, V., and Crawford, A. Internet paradox revisited. *Journal of Social Issues*. 58: 49-74, 2002.

[4] Malin, B., and Sweeney, L. *Compromising privacy in distributed population-based databases with trail matching: A DNA Example*. Tech Report CMU-CS-02-189. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. Dec 2002.

[5] Malin, B., and Sweeney, L.. Re-identification of DNA through an automated process. In *Proc American Medical Informatics Association Annual Symposium*, Washington, DC, pp 423-427, Nov 2001.

[6] Newcombe, H.B., Kennedy, J.M., Axford, S.J., and James, A.P. Automatic linkage of vital records. *Science*. 130: 954-959, 1959.

[7] State of Illinois Health Care Cost Containment Council. *Data release overview*. Springfield, 1998.Sweeney, L. Guaranteeing anonymity when sharing medical data, the Datafly system. In *Proc American Medical Informatics Association Annual Symposium*, pp 51-55, Washington, DC, Nov 1997.

[8] Sweeney, L. Uniqueness of simple demographics in the U.S. population. LIDAP-WP4. Laboratory for International Data Privacy, Carnegie Mellon University. 2000.

[9] Sweeney, L. Information explosion. *Confidentiality, disclosure, and data access: theory and practical applications for statistical agencies*, L. Zayatz, P. Doyle, J. Theeuwes and J. Lane (eds), Urban Institute, DC, 2001.

[10] Sweeney, L. K-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*. 10(7): 557-570, 2002.

[11] Torra, V. Re-identifying individuals using OWA operators. In *Proceedings of the 6th International Conference on Soft Computing*. Iizuka, Fukuoka, Japan. 2000.

[12] Winkler, W.E. Matching and record linkage. *In Business Survey Methods*, New York, J. Wiley. pp.355-384, 1995.