Secure Computation of k-Anonymous Distributed Data

Bradley Malin
Data Privacy Laboratory, Institute for Software Research International

May 2004
CMU-ISRI-04-120

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In a distributed environment, such as the World Wide Web, an individual leaves behind personal data at many different locations. To protect the privacy of an individual's sensitive information, locations make separate releases of identifiable data (*e.g.* name or social security number), and sensitive data (*e.g.* visitor's IP address). To the releasing location the data appears unlinkable, however, links can be established when multiple locations' releases are available. This problem, known as trail re-identification, manifests when an individual's location-visit patterns are reconstructed from, and linked between, sensitive and identifiable releases. In this paper, we present a protocol that enables locations to prevent trail re-identification without revealing identified or sensitive data. Instead, locations communicate encrypted versions of their datasets, such that decrypted data is never revealed until completion of the protocol. Via the protocol, every piece of sensitive data, released from any set of locations, is guaranteed to be equally relatable to at least k identities, or is k-anonymous.

Keywords: privacy, confidentiality, security, re-identification, k-anonymity, multiparty computation, quasi-commutative encryption, privacy protocols

1 Introduction

As technologies for collecting information infiltrate society, the ability to record and store personal information about specific individuals continues toward ubiquity. Knowingly and unknowingly, individuals shed data to a number of data collectors both within, as well as beyond, the confines of one's home. The information collection can be overt and apparent to the individual, such as when a consumer visits a retail store and completes a purchase with a personal credit card. Or data gathering can be less discernable, as when an individual's image is captured by an unforeseen video surveillance system. [1] Regardless, one thing is for certain; the collection of personal information is becoming more widespread. [2]

This is particularly the state of affairs within environments where an individual can leave related, and even the same, personal information behind at many different locations. For instance, in the realm of the World Wide Web, electronic commerce has facilitated the sharing and collection of personal information to an increasing number of independently functioning e-businesses. [3] Within this environment, websites collect differing types of data on individuals. Following the definitions of many online privacy policies, data is grossly categorized as *non-identifiable* and *identifiable* information. Non-identifiable information is information that does not explicitly reveal the identity of the individual. It is usually the case that an individual has little or no control over such information. By this definition, and as stated in many policies, the IP address of an individual's computer is considered non-identifiable information. An individual has no control over whether or not a website collects and stores their computer's IP address in the website's access log. In contrast, when visiting, a website an individual does have a choice regarding whether or not to share identifiable information. This type of information explicitly reveals the identity of the individual, such as name, residential address, or credit card number.

Data collectors relate identifiable to non-identifiable information for a number of legitimate purposes in accordance with their privacy practices. These purposes include direct marketing, website personalization, and fraud detection. To facilitate an individual's choice, websites post their privacy policies, which correspond to general aspects about how an individual's data will be used, managed, and shared. Thus, if an individual believes that a website's privacy practices are in accordance with their own, they may choose to provide their identifiable data. When an individual feels otherwise, they will not reveal identifying information. In this latter case, these individuals do not want such websites to know what name, or other identifiable information, corresponds to the visiting IP address.

Oftentimes, websites treat collections of person-specific, as well as access logs, information akin to commodities. The collected data can be shared, licensed, or sold with other parties for various purposes beyond in-house uses. Continuing with our example of e-commerce, customer lists are routinely provided to affiliated third parties. In the online environment, and beyond, it has been recognized that certain types of collected information about an individual are more sensitive than others. As specified in many privacy policies, non-identifiable data can not be shared in a manner that allows for it to be related to identifiable data. To account for this, many locations separate identifiable from non-identifiable data, and release the two as different datasets. This model of privacy protection appears to protect the identity of the individual. Releasing a list of IP addresses provides no more information than any other website might collect. There is no information that a data user, or adversary, can employ to re-identify the individuals of the released dataset. Right?

From the perspective of each data releasing location, the partitioning of non-identifiable and identifiable data appears to protect their consumers' privacy. However, exogenous factors infringe upon certain privacy protections that partitioning affords the data. Specifically, there are two factors that in combination rescind an individual's privacy. The first factor is the independent nature of data collecting locations. In many cases, it is a business advantage, or specified by the law, for a data collector to reveal data about consumers to a affiliated third party only. Data collectors rarely communicate information about their data collections to each other, but instead release data collections independently. The second factor is that individuals are not required, nor restricted to, visiting a particular data collecting location. Subsequently, data corresponding to each individual in a population can be collected and released by a different set of data collectors. As the number of locations that are collecting and sharing data increases, the location-visit patterns, or trails, of an individual's identifiable and non-identifiable information tends toward uniqueness. When an individual's identifiable and non-identifiable information can be uniquely matched to each other, sensitive information is re-identified! This problem, introduced by Malin and Sweeney [10, 11], is known as trail re-identification.

Trail re-identification is a real threat to the privacy of individuals whose data is distributed over multiple locations. Due to the recent formalization of the problem, there has been little research into methods for protecting against trail re-identification. In this paper, we present a protocol that allows for locations to work together to prevent trail re-identification without revealing their datasets to each other. The protocol guarantees that the trails constructed from any set of locations' data will adhere to k-anonymity protection. [4], [5] By this protection, the protocol guarantees that each piece of non-identifiable data will be equally relatable to a minimum of k identities via location-visit patterns. This research is the first to provide formal methods for preventing trail re-identification.

The remainder of this paper is organized as follows. In Section 2, relevant background to the trail re-identification problem is formally characterized and reviewed. In Section 3, relevant concepts from multiparty computation and encryption for the protection protocol are presented. Following the background, in Section 4, the protection protocol. In addition, particularly relevant security and privacy enabling characteristics are proven. In Section 5, security concerns with respect to the protocol are addressed. Finally, in Section 6, limitations and possible extensions to this work are discussed.

2 Trail Re-identification

In this section, a formal characterization of the models and data structures used throughout this paper is developed. The notation is based upon matrix algebra, and each location l maintains a dataset D_l as an $n \times m$ matrix. The columns of D_l are a set of semantically-defined attributes $A_l = A_{l1}, A_{l2}, \ldots, A_{lm}$. Each row vector corresponds to information about a single individual over the attributes. For example, $D_l[a_{i1}, \ldots, a_{in}]$ represents the values $a_{i1} \in A_{l1}, \ldots, a_{im} \in A_{lm}$ for the i^{th} row of dataset D.

D			
D^+			D^-
Name	**Residential Address**	**Item Purchased**	**IP Address**
John Doe	1 No Way	Book	128.2.98.4
Bob Smith	4 Some Place	Great Book	91.5.82.13
Mary Lamb	123 My Street	Not So Great Book	25.66.21.254

Figure 1: Sample dataset D_l with attribute set $A_l = \{$*Name*, *Residential Address*, *Purchase*, *IP Address*$\}$. The first two attributes are considered identifiable attributes, while the fourth attribute is considered non-identifiable.

A location's releases of non-identifiable and identifiable data are represented as submatrices of the matrix D_l. The submatrices are the result of a vertical-partitioning of D_l. We refer to a partition p as D_l^p. The first partition, called the de-identified submatrix D_l^-, is devoid of explicit identifiers. The second partition, called the identified submatrix D_l^+, contains explicit identifiers. To make evident the disjoint relationship between identifiable and non-identifiable data, the intersection of the submatrices is null[1], or $A_l^- \cap A_l^+ = \emptyset$. Fig.

[1]We neglect the fact that many IP addresses can leak geographic information. For more information on this topic, readers are

1 provides an example of a data collection $D_l\{$*Name*, *Residence*, *Item Purchased*, *IP Address*$\}$. A vertical partitioning of D_l is $D_l^+\{$*Name*, *Residence Item Purchased*$\}$ and $D_l^-\{$*IP Address*$\}$.

The goal of vertical-partitioning is to prevent an adversary from correctly reconstructing D_l. Therefore, while D_l^+ and D_l^- are submatrices of D_l, the ordering of the rows do not have to be equal to D_l. Once partitioning is performed, rows in the submatrices are randomly ordered. Though we omit the proof, it should be relatively simple to discern that the probability an adversary reconstructs D_l from D_l^- and D_l^+ alone is no better than random guessing. Thus, the relationship between every piece of data from D_l^+ and D_l^- is equivalent to the maximum degree of unlinkability as defined by Steinbacker and Kopsell. [9]

Though a location's de-identified submatrix is not susceptible to re-identification by itself, the susceptibility increases as more locations' releases are considered. Malin and Sweeney [10, 11] introduced a formal model of re-identification, referred to as trail re-identification, for an environment where multiple locations release data. It is this model of re-identification that the protocol below explicitly protects against. The model makes the assumption that an individual's identifiable and de-identified sensitive information is traceable across locations. Formally, this means that for every pair of locations l_i, l_j, there exists a set of relations between the attributes A_i^-, A_j^- and A_i^+, A_j^+. When these relations are sufficiently strong, then an individual's data can be traced across locations. For example, if the identified attributes $A_i^+ = A_j^+ = \{$*first name*, *last name*, *date of birth*$\}$, then it is assumed sufficient information exists to trace an individual from D_i^+ to D_j^+.

When an individual's data is traceable, the trail re-identification attack proceeds as follows. Let L be the set of data releasing locations. When the releases from these locations are collected by a single data collector, the released datasets are transformed into two location-based matrices. The first matrix is called the de-identified matrix N and it consists of sensitive information from the set of released datasets $D_1^-, \ldots, D_L^-$. The other matrix, called the identified matrix P, consists of identifiable information from the set of released datasets $D_1^+, \ldots, D_L^+$. We continue with the construction of N. The construction of P is a simple corollary. The dimensions of N are (the number of distinct data pieces) $\times$ ($|\bigcup_{l \in L} A_l^-| + |L|$). The first $|\bigcup_{l \in L} A_l^-|$ columns correspond to the released sensitive data. The latter $|L|$ attributes correspond to location-based information. Without loss of generality, we assume that the attribute set is the same for each location's release and the number of columns is $|A^-|$+$|L|$. For the x^{th} row vector in matrix X, the latter $|L|$ attributes are referred to as the trail of the data, or $trail(x, \mathsf{X})$. Furthermore, we use $trail(l, x, \mathsf{X})$to refer to the value of the l^{th} cell in $trail(x, \mathsf{X})$. We refer to the identity or de-identity from a row vector as $identity(x, \mathsf{X})$ and $deidentity(x, \mathsf{X})$, respectively.

Re-identification of $deidentity(n, \mathsf{N})$ to $deidentity(p, \mathsf{P})$ occurs when $trail(n,\mathsf{N})$ is correctly matched with $trail(n, \mathsf{P})$. Malin and Sweeney provide several re-identification algorithms, collectively termed REIDIT (re-identification of Data in Trails) for location-based attributes with Boolean values, where 1 and 0 represent the presence and absence of information at a location, respectively. [12, 10, 11] The REIDIT algorithms correctly link rows of N to P by exploiting unique patterns in the trails.

In addition, the effects of data completeness are taken into account. A matrix X is said to be *unreserved* to a matrix Y, if for every individual, the data trails corresponding to the individual in both matrices are equivalent. In some situations, an individual leaves behind both identifiable and de-identified data to a location. However, there are times when a location does not release all data that it has in its possession. In either of these cases, matrix X is said to be *reserved* to matrix Y if the trail of each individual in matrix X, $trail(x, \mathsf{X})$ can be transformed into the individual's corresponding $trail(y, \mathsf{Y})$ in matrix Y by flipping only Boolean values of 0 to 1. When this transformation can be performed, we say that The relationship these trails is such that $trail(x, \mathsf{X})$ is a subtrail (represented with the $\preceq$ symbol) of $trail(y, \mathsf{Y})$. Similarly, $trail(y, \mathsf{Y})$ is said to be the supertrail of $trail(x, \mathsf{X})$, or $trail(y, \mathsf{Y}) \succeq trail(x, \mathsf{X})$. It is this more gen-

directed to consult [6], [7], and [8].

Algorithm 1 REIDIT-I (X, Y)

$\{$***Assumes:*** *1)* X *and* Y *consist of de-identified and identified data, respectively*; *2)* X *is reserved to* $\mathsf{Y}\}$
$REID \Leftarrow \emptyset$
for $n = 1$ to $|\mathsf{X}|$ **do**
  if there is one and only one y, such that $trail(n, \mathsf{X}) \preceq \mathrm{trail}(\mathrm{y}, \mathsf{Y})$ **then**
    $REID \Leftarrow \langle identity(y, \mathsf{Y}), deidentity(n, \mathsf{X})\rangle \cup$ REIDIT-I(X- n, Y- y) $\{$*Remove n and y from further consideration*$\}$
  end if
end for
if $|\mathsf{X}| \equiv |\mathsf{Y}|$ **then**
  for $m = 1$ to $|\mathsf{Y}|$ **do**
    if there is one and only one x, such that $trail(m, \mathsf{Y}) \succeq \mathrm{trail}(\mathrm{x}, \mathsf{X})$ **then**
      $REID \Leftarrow \langle identity(m, \mathsf{Y}), deidentity(x, \mathsf{X})\rangle \cup$ REIDIT-I(X- x, Y- m) $\{$*Remove x and m from further consideration*$\}$
    end if
  end for
end if
return $REID$

eral scenario that we consider for this research. Fig. 2 provides an example of location-based matrices where matrix P is reserved to matrix N. In this example, $trail(Mary, \mathsf{N}) \preceq \mathrm{trail}(167.92.182.1, \mathsf{P})$ and $trail(Mary, \mathsf{N}) \preceq trail($*114.32.40.81*,$\mathsf{P})$.

D_1^+

Name
John
Mary

D_1^-

IP
128.2.41.234
167.92.182.1
114.32.70.81

D_2^+

Name
Bob
Kate

D_2^-

IP
128.2.41.234
32.221.5.15
167.92.182.1

D_3^+

Name
John
Mary
Kate

D_3^-

IP
32.221.5.15
167.92.182.1
114.32.70.81

D_4^+

Name
Bob
John

D_4^-

IP
32.221.5.15
128.2.41.234
114.32.70.81

P

Name	l_1	l_2	l_3	l_4
John	1	0	1	1
Mary	1	0	1	0
Bob	0	1	0	1
Kate	0	1	1	0

N

IP	l_1	l_2	l_3	l_4
128.2.41.234	1	1	0	1
167.92.182.1	1	1	1	0
32.221.5.15	0	1	1	1
114.32.70.81	1	0	1	1

Figure 2: (left)Releases of four locations, (right)Resulting location-based matrices

The REIDIT-Incomplete (REIDIT-I) algorithm, pseudocode of which is provided, returns correct re-identifications. No false re-identifications are made. This algorithm will be used to validate our protection protocol. The algorithm works as follows. 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 (Proof provided in [10]). The re-identified trails from N and from P are removed. Processing continues until no more re-identifications can be made.

3 Quasi-commutative Encryption

The protection protocol described below makes use of an interesting concept from secure multiparty computation known as the one way accumulator, or OWA. [13] In previous research, OWAs have been applied to a variety of distributed secure computations. For example, Zachary [14] demonstrates that OWAs provide the necessary features for securely testing membership of nodes in distributed sensor networks. Faldella and Prandini [15] make use of OWAs for certificate authentication in a distributed public-key infrastructure. The protocol also employs OWAs for computation in a distributed environment. However, with respect to this research, one think of an OWA as a method that empowers disparate locations, using different encryption keys, with the ability to reveal encrypted information from their local datasets, such that an encrypted identity or de-identity is equivalent across locations. The OWA applied in this manner permits trail re-identification and protection methods to be computed over encrypted data. Plaintext information need not be revealed until it is computationally guaranteed that re-identification is impossible.

First, we review the general concepts of OWAs, then their transformation into a blinding cryptosystems. An OWA is a hash function $h : X \times Y \rightarrow X$ that satisfies the *quasi-commutative* property. In equation (1), the following property holds for an arbitrary number and ordering of y_i.

$$h(h(x, y_1), y_2) = h(h(x, y_2), y_1) \tag{1}$$

Benaloh and de Mare note that the modular exponentiation function $e_n(x, y_i) = x^{y_i} mod(n)$, as defined in RSA encryption, is an OWA. [16] For appropriately chosen n, where n is the product of two large prime integers p, q, computing x from $e_n(x, y_i)$ and y can not be accomplished in polynomial time. Since repeated use of e_n may reveal hash collisions, values of n are further restricted to be chosen from the set of *rigid integers*, where n is the product of two safe primes p, q. A prime number p is a safe prime if $p = 2p' + 1$, where p' is an odd prime.

The trapdoor feature of modular exponentiation was exploited by Kantarcioglu and Clifton [17], such that OWAs can be converted into public key cryptosystems. In order to do so, each encryption key y_i is paired with a decryption key z_i, where $y_i * z_i = 1 mod(\varphi(n))$.[2] When y_i and z_i are defined in this manner, decryption of an encrypted value v can proceed over m independent locations as

$$x = (h \ldots h(h(v, z_1), z_2), \ldots z_m) \tag{2}$$

Again, the ordering of the decryption keys $z_1, z_2, \ldots, z_m$ is of no consequence. Thus, the encrypted value v can be decrypted in a sequential manner using the same hash function as $h(x, z_i) = x^{z_i} mod(n)$.

4 Protection Protocol

In this section, we construct a protocol that explicitly prevents trail re-identification. The protocol is called *central authority trail anonymization*, or the CATA protocol. As the name implies, the current implementation requires a central authority. We assume that the central party is semi-trusted. It is trusted to receive and analyze encrypted data, but not plaintext data. This central party will be permitted collect encrypted data from each of the data releasing locations. In addition, given encrypted data, the central authority is expected to return honest information to each location.

4.1 CATA Protocol

We begin with a general overview of the protocol. A more in-depth description and formal treatment follows. First, each location encrypts every other location's de-identified and identified submatrices. Then, the central authority is provided with the encrypted datasets. Upon reception of each locations dataset, the central authority runs its own trail re-identification and anonymization techniques. Discovered re-identifications inform the central party which encrypted values would be re-identified if they were released in their plaintext

[2]The term $\varphi(n)$, Euler's totient function, specifies the number of relatively prime positive integers less than n.

form. Given the re-identifications, the central party determines which encrypted data must be removed by each location in order to anonymize the trails. A de-identity, such as an IP address, is considered anonymous if its corresponding trail can not be correctly matched to its identity. We say that a trail from matrix X is k-anonymous if the trail is equally relatable to k trails from matrix Y . Thus, we will use k as a protection parameter. The greater the value of k, the more protection is afforded to the data. Once the central party has sufficiently anonymized the data, it returns a list of encrypted values to each location. The encrypted values are decrypted by the set of locations, such that the final decrypter is the location the list was destined for. The decrypted data is removed from the locations' releases. Finally, when the locations datasets have been reduced as specified by the central party's lists, each location releases its plaintext datasets.

More formally, the CATA protocol is defined as follows. There are two types of participants, data releasing locations $L = \{l_1, l_2, \ldots, l_{|L|}\}$ and a central authority C. Each location $l \in L$ maintains a pair of encryption and decryption keys, $<y_i, z_i>$, for a reversible quasi-commutative encryption function h as defined above. These keys are kept private, akin to Chaum's blinding system. [18] The encryption function is known to all data releasing parties. We now step through the protocol.

Step 0a: Participating Location. (Partitioning) Prior to releasing any data, each location l partitions its data collection matrix D_l into a de-identified and identified submatrix, D_l^- and D_l^+, respectively.

Step 0b: Central Party. (Path Allocation) The central party issues an encryption path p_l^e and decryption path p_l^d for each location l. There are certain constraints on the paths that can be issued, which will be discuss below in the security analysis.

The following Steps 1 and 2 are equivalent for D_l^- and D_l^+. Without loss of generality, we continue with the encryption process for D_l^- only.

Step 1. (Initial Encryption) Each location l encrypts each value in D_l^- using y_l and h. For simplicity, we represent the set of encrypted values as $h(D_l^-, y_l)$. After initial encryption, a hashed dataset $h(D_l^-, y_l)$ exists for, and is in the possession of, each location.

Step 2. (Full Encryption) After a location encrypts its dataset, it shuffles the ordering of the rows in $h(D_l^-, y_l)$ and sends it to the next location in path p_l^e for encryption. This process continues in a sequential manner, for each dataset, until every location has hashed the dataset with its own encryption key. For D_l, we say that the dataset is *full encrypted* when every location in L has encrypted it. We refer to the full encrypted dataset as $f(D_l^-) = h(h \ldots (h(h(h(D_l^-, y_l), y_1), y_2), \ldots, y_{|L|-1}), y_{|L|})$.

Step 3. (Encrypted Re-identification) Once a dataset is full encrypted, the final encrypter submits the it to the central authority C. Upon receiving all full encrypted datasets, the C constructs de-identified N and identified P location-based trail matrices. At this point, C performs re-identification and reduction over the location-based matrices. Reduction is performed, such that the resulting matrices adhere to the k-anonymity formal protection model. [5] By adhering to this model, our method guarantees that for any element in a released dataset there are k-1 other elements that are indistinguishable from that element over some distinguishability function. For our model, an element is a data trail and the distinguishability function is the REIDIT-I algorithm. The current implementation of the protocol uses a simple method we call the *Random k-Obscure* algorithm, the pseudocode of which is provided below.

The Random k-Obscure algorithm accepts three parameters; the first two consists of the location based matrices and the third is the anonymity parameter k. To begin, the algorithm computes the minimal distance

Algorithm 2 Random k-Obscure (N, P, k)

$\{$***Assumes:*** P is reserved to $\mathsf{N}\}$

Let X be a $|\mathsf{P}| \times |\mathsf{N}|$ matrix, where $X[x_{pn}]$ equals the minimum number of Boolean location values in trail $trail(p, \mathsf{P})$ for $trail(p, \mathsf{P}) \preceq trail(n, \mathsf{N})$, to be true

Let $S = \{s_1, s_2, \ldots, s_{|L|}\}$ be a set of $|L|$ empty lists

for $i = 1$ to $|\mathsf{P}|$ **do**

  Let r_i equal the number of cells equal to zero in the i_{th} row of X

  if $r_i < k$ **then**

    Let Z be the set of $k - r_i$ indices of row i in X with the smallest values > 0

    for each $z \in Z$ **do**

      Let B be the set of indices where $trail(b, i, \mathsf{P}) \succeq trail(b, z, \mathsf{N})$

      for each $b \in B$ **do**

        $trail(b, p, \mathsf{P}) \Leftarrow 0$

        $s_b \Leftarrow s_b \cup identity(p, \mathsf{P})$

      end for

    end for

  end if

end for

if $|\mathsf{N}| \equiv |\mathsf{P}|$ **then**

  for $i = 1$ to $|\mathsf{N}|$ **do**

    Let c_i equal the number of cells equal to zero in the i_{th} column of X

    if $c_i < k$ **then**

      Let Z be the set of $k - c_i$ indices of column i in X with the smallest values > 0

      for each $z \in Z$ **do**

        Let B be the set of indices where $trail(b, z, \mathsf{P}) \succeq trail(b, i, \mathsf{N})$

        for each $b \in B$ **do**

          $trail(b, p, \mathsf{P}) \Leftarrow 0$

          $s_b \Leftarrow s_b \cup identity(p, \mathsf{P})$

        end for

      end for

    end if

  end for

end if

return S

necessary to convert trails of identifiable data into subtrails of de-identified data. Let r_t be the number of trails that trail t is a subtrail of. If t is the subtrail of at least k pieces of data (i.e. $r_t \geq k$), t is sufficiently protected. Otherwise, the method finds the $k - r_t$ trails of closest distance to t. The bits of value 1 that convert t into a subtrail of the r_t trails are flipped to value 0. For every bit flip, the trail value is allocated to the appropriate list $s_1, \ldots, s_{|L|}$ to be returned to locations $l_1, \ldots, l_{|L|}$. Each value in a list s_l is an encrypted value that location l must remove from D_l^- to prevent re-identification.

Step 4. (Full Decryption) Each list s_l is sent back to L for decryption via path p_l^d. When l decrypts s_l, the s_l is said to be full decrypted. The decryption of the dataset proceeds sequentially. Thus, the full decryption can be represented as $f(s_l)$:

$$h(h \ldots h(h \ldots (h(h(s_l, y_1), y_2), \ldots, y_{l-1}), y_{l+1}), \ldots, y_{|L|}), y_l).$$

Step 5. (Local Obscure) At this point, each location l is in possession of $f(s_l)$, a plaintext listing of entries from D_l^-. For location l to ensure that k-anonymous privacy, they must remove all entries in $f(s_l)$ from D_l^-. Each l reduces its dataset and releases the $(D_l^- - f(s_l))$ and D_l^+.

4.2 Protocol Example

For a more concrete understanding of the protocol, we will walk through an example. Consider the datasets from Fig. 1 with the following numerical representations substituted for names and IP address. For names, let John = 100, Mary = 200, Bob = 300, and Kate = 400. Similarly, let their corresponding IP address be 128.2.41.234 = 1000, 167.92.182.1 = 2000, 32.221.5.15 = 3000, and 114.32.70.81 = 4000. Using this mapping, D_1^+ = [100, 200] and D_1^- = [1000, 2000, 3000].

Let $n = 11 * 839 = 9229$, $h(x, y) = x^y mod(n)$, and the set of encryption and decryption key pairs $\langle y_i, z_i \rangle$ be {<31, 811>, <199, 379>, <227, 443>, <337, 373>}. For each of the locations, C generates a random path for each dataset to follow for serialized encryption. Let the set of paths be $\{\langle l_1 \rightarrow l_2 \rightarrow l_4 \rightarrow l_3 \rangle, \langle l_2 \rightarrow l_4 \rightarrow l_3 \rightarrow l_1 \rangle, \langle l_3 \rightarrow l_1 \rightarrow l_4 \rightarrow l_2 \rangle, \langle l_4 \rightarrow l_3 \rightarrow l_1 \rightarrow l_2 \rangle\}$. The serialized full encryption for D_1^+ is h(h(h(h([100, 200], 31), 199), 337), 227) = [3004, 2191]. After C receives the all full encrypted datasets, it constructs the location based matrices over the encrypted values and runs the random k-obscure algorithm with $k = 2$. This level of protection has already been achieved for all values, except for 3004. The closest de-identified trail to 3004 is the trail of 3277, which is of distance 0. Thus, one more trail is necessary to satisfy the k protection level.

Since all other de-identified trails are equidistant with a distance of 1 from 3277, we randomly choose a trail, say 3990, to make 3004 a subtrail of. The only index that needs a bit flip is index 1. So, this bit is flipped into 0. The set of information to return to the participating locations is S={{3004}, {}, {}, {}}. Again, C generates random paths for the encrypted datasets to follow to their locations. Here, since we only need one path, let this path be $\langle l_2 \rightarrow l_4 \rightarrow l_3 \rightarrow l_1 \rangle$. The decryption of s_1 proceeds as $h(h(h(h($[3004], 379), 373), 443), 811) = [100].

4.3 Correctness

The CATA protocol allows for claims about the privacy and security of the data to be made. Here, we prove several crucial guarantees as theorems. For now, we assume that there exists no collusion among parties. In the following section, the effects of collusion and practical ways minimize the effects of such, are discussed.

The first aspect of CATA that we prove is its ability to prevent an independent location from learning the plaintext information of encrypted data. Let D_i^p be an arbitrary partition of D_i.

Theorem 1. There exists no location l_j, where $l_j \neq l_i$, that can independently determine D_i^p from $h(D_i^p, y_i)$.

Proof. Let $h_x(D_i^p)$ be the values of D_i^p hashed by an arbitrary number of x locations as observed by l_j. In the base case, which is the best scenario for l_j, the dataset has been hashed by one location only, $h_x(D_i')$ = $h_1(D_i')$. If $x = 1$, then D_i^p must have been hashed by l_i only: $h_1(D_i')$ = $h(D_i, y_i)$. Obviously, since l_j does not know either y_i or z_i, it can not determine the plaintext value for any of the encrypted values.

However, we must also account for the additional manipulation that l_j is capable of; l_j can hash the dataset with its own key to create $h_2(D_i^p)$ = $h(h(D_i, y_i), y_j)$. This new dataset can reveal much, that is, if l_j could encrypt its own dataset as $h_2(D_j^p)$ = $h(h(D_j, y_j), y_i)$. In this case, the intersection of the datasets, $h_2(D_i^p) \cap h_2(D_j^p)$, reveals encrypted values from l_i's dataset that l_j knows the decrypted values of - because they exist in l_j's dataset as well. Yet, this is impossible, since l_j can never recover $h_2(D_j^p)$. Under the CATA protocol, no location receives their dataset as hashed by another location. The only values of l_j's dataset that l_j knows are the plaintext values D_j, the hashed values D_j, and the hashed values $h(D_j, y_j)$. □

Now that simple security has been established, we concentrate on the privacy of the released datasets. Let L be the set of participating locations participating in the CATA protocol. Let N and P be location-based matrices constructed from reduced datasets from all locations in L, with P reserved to N.

Theorem 2. There exists no $identity(p, \mathsf{P})$ released from a subset of locations $L' \subseteq L$ that can be re-identified to its corresponding $deidentity(n, \mathsf{N})$ with probability $\leq 1/k$.

Proof. In the base case, we consider the set of encrypted data releases from all locations in L and the distance matrix X as defined in the k-obscure algorithm above. For an arbitrary $trail(p, \mathsf{P})$, the rowsum of X corresponds to the number of supertrails for $trail(p, \mathsf{P})$ in N. If the rowsum of X for $trail(p, \mathsf{P})$ is $< k$, then bits of value 1 in $trail(p, \mathsf{P})$ are flipped until the number of supertrails for $trail(p, \mathsf{P})$ equals k. Thus, the probability that an adversary could map the plaintext $trail(p, \mathsf{P})$ from the set of reduced datasets of $|L|$ locations is at most $1/k$. The probability is $\leq 1/k$, and not equal to $1/k$, because there it is possible that the flipping of bits in $trail(p, \mathsf{P})$ has created supertrails of $trail(p, \mathsf{P})$ in Nthat are beyond the set of k trails in Nconsidered for reduction of $trail(p, \mathsf{P})$.

In the more general case, when the set of releasing locations is $L' \subseteq L$, the probability of re-identification for an arbitrary $trail(p, \mathsf{P})$ remains no better than it was in the base case. Let $\mathsf{N}_{\mathsf{L}'}$ and $\mathsf{P}_{\mathsf{L}'}$ be the location matrices generated by encrypted data from the data releases of L'. Now, let $trail(p, \mathsf{P}')$ be the subtrail of at least k trails in N as guaranteed by random k-obscure. Each $trail(p, \mathsf{P}')$ is equivalent to $trail(p, \mathsf{P})$ when P is constructed from locations $L \cap L'$. This is the same as making $trail(p, \mathsf{P}') \preceq trail(p, \mathsf{P})$ by zeroing out all bits of indices $L \cap L'$. The number of supertrails for $trail(p, \mathsf{P})$ with the values of indices $L \cap L'$ set to 0, must be y, where $y \geq k$. Furthermore, zeroing out the values for the indices of every trail in N makes the number of supertrails for $trail(p, \mathsf{P})$ with the values of indices $L \cap L'$ set to 0, equal to y. Since trails from the zeroed out P and N matrices is the same as using P' and N', the probability that any $trail(p, \mathsf{P}')$ can be re-identified must be $1/y$. And since $y \leq k$, the probability that a correct re-identification is made is $\leq 1/k$.□

4.4 Computational Overhead

Assume that encryption and decryption can be done in constant time. Encryption of each dataset is distributed across locations and the total number of encrypts performed by any location is $|L|$. In addition, each location must make $|L| + 1$ communications, the first $|L|$ to pass encrypted datasets to the next location and the final step for submission to the central party. The order of complexity is the same for decryption of datasets returned from the central party. Thus the computational overhead for the participating locations is due to encryption and decryption, and can be performed in O($|L|$) time with O($|L|^2$) communication messages.

The majority of computation is the burden of the central location. Assuming that P is reserved to N, $|\mathsf{P}| \leq |\mathsf{N}|$, so in worst case $|\mathsf{P}| = |\mathsf{N}|$. Thus, construction of the location based matrices can be completed in O($|\mathsf{N}|^2$). The construction of the trail distance matrix can be completed by first by sorting in O($|L||\mathsf{N}| \log |\mathsf{N}|$). Conversion of each trails in P the subtrails of k trails in N can be completed in O($k|N||L|$). Therefore, the order of complexity for the central location is O($|\mathsf{N}|^2$) + O($|L||\mathsf{N}| \log |\mathsf{N}|$) + O($k|N||L|$). When $|L| < |N|$, complexity is approximately O($|\mathsf{N}|^2$). When $|N| < |L|$, $k < |L|$, and complexity is approximately O($|L|$). Finally, when $|N| \approx |L|$, complexity is approximately O($|L||\mathsf{N}| \log |\mathsf{N}|$). In the real world, it is expected $|N| \gg |L|$.

5 Security Concerns

Theorem 1 only guarantees security when each location functions independently. Though a single location can not independently discern any plaintext values of another location, colluding locations can collaborate to bound the set of plaintext values an encrypted value corresponds to. Colluding locations can not discern the exact plaintext values due to the fact that non-colluding location will perform random shuffling of hashed datasets. Let L be the set of participating locations and L. For example, referring back to the protocol example, if l_1 knows that the full encrypted value 2191 resides in both $f(D_1^+)$ and $f(D_3^+)$, then l_1 learns that l_3 has either John or Mary in D_1^+.

The are several types of collusion that can exist, which we now explore.

5.1 Central Party Collusion

When collusion occurs between participating locations and the central party, each colluding locations can compare their full encrypted dataset with full encrypted data set of another party. As stated above, if an encrypted value v is found to be common between the colluding and non-colluding locations full encrypted data sets, then the colluding party can bound the set of plaintext values for v. When there are multiple colluding parties, it is possible that the exact plaintext value for v can be learned. This would occur when $trail(v, \mathsf{V})$ is unique over the set of colluding locations indices. When the trail of v is unique, the plaintext value of v is uniquely determined by mapping to the lone value resulting from the union of the colluding paries datasets.

5.2 Non-random Data Paths

If the allocation of paths for dataset encryption and decryption are chosen at random, then the following type of collusion can occur. Let $P = p_1, \ldots, p_{|L|}$ be the set of paths for locations $l_1 \ldots l_{|L|}$. Imagine a scenario with two colluding locations l_i and l_j and non-colluding location l_k. Let x_i^{ij} be the set locations that l_i's dataset passes through between, and including, l_i and l_j in p_i. There are several ways that colluding locations can bound l_k's values. Collusion can occur when any of the following conditions are satisfied

$$x_i^{ij} - l_j = x_k^{ki} \tag{3}$$

$$x_i^{ij} = x_k^{ki} \tag{4}$$

$$x_i^{ij} = x_k^{kj} \tag{5}$$

When Equation 3 holds, l_j receives l_i's dataset after both l_i and l_k have hashed it. Also, l_i receives l_k's dataset after it has been hashed by the same set of locations that hashed its own dataset. Consider the following paths $p_i = \langle\, l_i \rightarrow l_c \rightarrow l_k \rightarrow l_j \rightarrow l_a \rangle$ and $p_k = \langle l_k \rightarrow l_c \rightarrow l_i \rightarrow l_a \rightarrow l_j \rangle$. For an arbitrary value v, location

l_j receives $h(h(h(v, y_i), y_c), y_k)$ and location l_i receives $h(h(h(v, y_k), y_c), y_i)$. By the definition of the hash function h, these two hashes are equivalent.

Equations 4 and 5 are mutually exclusive properties. They can not both be true at the same time because either $x_k^{ki} \subset x_k^{kj}$ or $x_k^{ki} \supset x_k^{kj}$ is true. When Equation 4 holds, l_j has l_i's dataset after it has been hashed by l_k; and l_i holds $l_k's$ dataset after it has been hashed by l_j. Equation 5 allows for the same, except now l_j has l_k's dataset. For an example that satisfies this condition, consider the following paths: p_i = $\langle l_i \rightarrow l_c \rightarrow l_k \rightarrow l_e \rightarrow l_j \rangle$ and p_k = $\langle l_k \rightarrow l_c \rightarrow l_i \rightarrow l_e \rightarrow l_j \rangle$. With these allocated paths, x_i^{ij} = x_k^{kj} = $\{l_c, l_e, l_i, l_j, l_k\}$ and for an arbitrary value v:

$$h(h(h(h(v, y_i), y_c), y_k), y_e), y_j) = h(h(h(h(v, y_k), y_c), y_i), y_e), y_j)$$

Based on this phenomena, it is evident that data paths should not be chosen at random. Rather, they should be chosen such that none of Equations 3 - 5 are satisfied. Path allocation in this manner would prevent collusion among all locations in L; however, there is a small problem. Path allocation that does not satisfy Equations 3 - 5 can be achieved for up to $|L| - 1$ colluders, but the central party must know which locations are colluding. For example, one simple way to prevent collusion is to generate paths, such that the first $|U|$ positions of a colluding locations path consists only of locations from U.

When the set of colluders is unknown to the central location proper path allocation is impossible to achieve. This can be illustrated with a simple contradiction proof. First, assume that such a set of paths exists that Equations 3 - 5 are never satisfied. Let $L = \{l_1, l_2, l_3\}$ and the set of colluders $U = \{l_2, l_3\}$. There is only one way to allocate l_1 in the three paths to prevent collusion. Obviously, l_1 must be in the first position of p_1. For paths p_2 and p_3, l_1 must be in the final position, or else one of the colluders could capture the other colluders dataset as hashed by l_1. Now, if we change the set of colluders to equal $U = \{l_1, l_2\}$, both Equation 3 and 5 are satisfied.

The ability to generate a set of paths that minimize collusion with an unknown U is easier to achieve as $|U|/|L|$ decreases. In future research, one of our goals is to determine methods for generating paths when the colluders are unknown.

6 Conclusions and Future Work

This work introduced a novel protocol, termed central authority trail anonymization (CATA), for anonymizing a set of individual's location visit patterns. It is the first protocol explicitly proven to prevent trail re-identification. The protocol allows for multiple locations to conduct distributed re-identification analysis before plaintext, as well as proprietary information ($e.g.$ the purchases made by individuals at a particular location). Though the protocol facilitates anonymization, one of the necessary areas for extension to this research is the design of more efficient anonymization schemas. While any number of possible methods could be employed to anonymize trails, one must ask when is one method better than another. For example, the random k-obscure generates 2-anonymous trail distance matrices for the trails in Fig. 3 with equal probability. Yet, there are several drawbacks to using such an algorithm that are opportunities for extending this research.

First, random k-obscure measures the distance between trails as a scalar. While a scalar is an unbiased metric for measuring the distance between two trails, it suppresses necessary information for relating the distance between three or more trails concurrently. It is possible that distance vectors (*i.e.* n-dimensional with one dimension for each index of a trail) will be more useful in optimizing the choice of which bits in a trail should be used for anonymization. [19]

Second, the randomness and greediness of the random k-obscure algorithm leaves much room for optimizing the obscuring method. This issue is particularly pertinent to situations when there are many trails

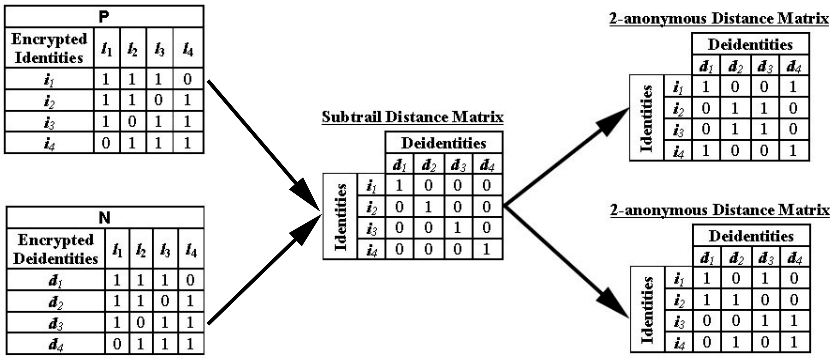

Figure 3: Variants of the 2-anonymous distance matrices generated via random k-obscure.

within an equal distance of each other, but only a subset are necessary to achieve k-anonymous trails. Which trails should be chosen? Optimization over a distance vector can help in this decision, but it be of great assistance if an objective function is considered. Notice that in Fig. 3, two different 2-anonymous trail distance matrices are shown. Each one has redeeming characteristics. The matrix in the top-right maximizes the number of pieces of data each location can release, whereas the matrix in the bottom-right maximizes variance in the relationships between trails.

Third, there is the definition of anonymity itself. The k-anonymity model defines anonymity through indistinguishability, or the ability to tell data apart. [4], [5] Initially applied to field- structured data, it has been extended and adapted for an ever widening field of data types, from anonymous message transmission [20] to privacy preserving facial recognition systems [21]. However, the current k-anonymity model, though computational, is deterministic in the characterization of anonymity. Other models characterize anonymity from a more probabilistic framework. For example, recent models of communication [22], [23] define anonymity in terms of information theory. In a sense, these models are very similar to k-anonymity. They both measure the amount of information that an adversary can use to distinguish between different identities. Yet, k-anonymizing data is not equivalent to stating that the probability of re-identification for the data is less than or equal to $1/k$ due to lack of confirmation. The adversary is only assured that, in a best case, he can relate k identities, one of which is correct, to the data. It will be interesting to see how anonymizing methods based on k-anonymity compare to these other types of models. Comparison and analysis into the relationships between information theoretic and k-anonymous models will help to further the definition of anonymity.

The proper choice of anonymization method will be dependent on the objective of the parties involved and will be dictated by the needs of the data users. Thus, the design of applied anonymization methods is both an interesting and challenging area of research for the computer science community.

Acknowledgements

The author wishes to thank the members of the Data Privacy Laboratory at Carnegie Mellon University for their support and encouragement. The author particulary acknowledges useful discussions and insight into this research provided by Edoardo Airoldi and Yiheng Li. This work was supported by the Data Privacy Laboratory at Carnegie Mellon University.

References

[1] M. Jones. All eyes are on Oceanfront's new surveillance system. *The Virginian-Pilot*. September 10, 2002.

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

[3] D. MacDonald and P. Higgins. E-business investment benchmarking study. A.T. Kearney and Line56 Research. August 2003. Available at: http://www.line56.com/research/download/L56_ATKearney_-Benchmarking_Research_0703.pdf

[4] P. Samarti and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. *Technical Report SRI-CSL-98-04*. Computer Science Laboratory, SRI International, 1998.

[5] L. Sweeney. k-Anonymity. a model for protecting privacy. International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems. Vol. 10, No. 5, 2002, pp. 557-570.

[6] M. Dodge and N. Shiode. Where on Earth is the Internet? An empirical investigation of the spatial patterns of internet "real-estate" in relation to geospace in the United Kingdom. In *Telecommunications and the City Conference*, Athens, GA, March 1998.

[7] O. Buyukkten, J. Cho, H. Garcia-Molina, L. Gravano, N. Shivakumar. Exploiting geographical location information of web pages. In *WebDB '99, with ACM SIGMOD*, 91-96, Philadelphia, PA, June 1999.

[8] B. Cheswick, H. Burch, and S. Branigan. Mapping and visualizing the Internet. In *USENIX 2000*, San Diego, CA, June 2000.

[9] S. Steinbrecher and S. Köpsell. Modelling unlinkability. In *Privacy Enhancing Technolgies Workshop (PET) 2003*, Dresden, Germany, March 2003.

[10] B. Malin and L. Sweeney. Compromising online anonymity with trail re-identification. *Data Privacy Laboratory Working Paper LIDAP-WP14*. School of Computer Science, Carnegie Mellon University. *Presented at Privacy in D.A.T.A. Workshop*, Pittsburgh, PA, March 2003. http://www.aladdin.cs.cmu.edu/workshops/privacy/index.html

[11] B. Malin and L. Sweeney. How (not) to protect privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. *Forthcoming in the Journal of Biomedical Informatics*. 2004. *A prior version is available as Tech Report CMU-ISRI-04-115, Carnegie Mellon University, Pittsburgh, PA, 15213, April 2004*.

[12] B. Malin and L. Sweeney. Re-identification of DNA through an automated linkage process. In *American Medical Informatics Annual Symposium (AMIA) 2001*, Washington, DC, 423-427, 2001.

[13] J. Benaloh and M. deMare. One-way accumulators: a decentralized alternative to digital signatures (Extended Abstract). In: Hellsuth, T. (ed.): Advances in Cryptology (EUROCRYPT '93). LNCS 765. Springer-Verlag New York 1994, pp. 274-285.

[14] J. Zachary. A decentralized approach to secure group membership testing in distributed sensor networks. In MILCOM 2003. Boston, MA, Oct. 2003.

[15] E. Faldella and M. Prandini. A novel approach to on-line status authentication of public-key certificates. In *16*th*Annual Computer Security Applications Conference*. New Orleans, LA, Dec. 2000.

[16] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, Vol. 21, No. 2, Feb. 1978, pp. 120-126.

[17] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed data mining of association rules on horizontally partitioned data. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD)*. Madison, WI, USA, June 2002.

[18] D. Chaum. Blind signatures for untraceable payments. Advances in Cryptography, Crypto 1982. Plenum Press. 1983: pp. 199-203.

[19] L. Sweeney. Computational disclosure control: a primer on data privacy protection. Ph.D. Thesis, Massachusets Institute of Technology, Cambridge, MA, 2001.

[20] L. von Ahn, A. Bortz, and N. Hopper. k-anonymous message transmission. In 10^{th} *ACM Conference on Computer and Communications Security*, Washington, DC, Nov. 2003, pp. 122-130.

[21] E. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying facial images. *Tech Report CMU-CS-03-168*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, June 2003.

[22] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In P. Syverson and R. Dingledine (eds.), Privacy Enhancing Technologies 2002, LNCS 2482, San Francisco, CA, April 2002, pp. 41-53.

[23] C. Diaz, S. Seys, J. Clawssens, and B. Preneel. Towards measuring anonymity. In P. Syverson and R. Dingledine (eds.), Privacy Enhancing Technologies 2002, LNCS 2482, San Francisco, CA, April 2002, pp. 54-68.