Rethinking Anonymity for Social Networks

Aaron Beach, Mike Gartrell, Richard Han
University of Colorado at Boulder
aaron.beach, mike.gartrell, richard.han@colorado.edu

Technical Report CU-CS 1065-10

June 2010

Rethinking Anonymity for Social Networks

Aaron Beach, Mike Gartrell, Richard Han
University of Colorado at Boulder
{aaron.beach, mike.gartrell, richard.han}@colorado.edu

***Abstract*—This paper explains why existing anonymity models such as k-anonymity cannot be applied to the most common form of private data release on the internet, social network APIs. An alternative anonymity model, PP-anonymity, is presented, which measures the posterior probability of an attacker logically deducing previously unknown private information using a social network API. Finally, the feasibility of such an approach is evaluated suggesting that a social network site such as Facebook could practically implement an anonymous API using PP-anonymity, providing its users with an anonymous option to the current application model.**

I. Introduction

Traditional anonymity research assumes that data is released as a research-style microdata set or statistical data set with well understood data types. Furthermore, it is assumed that the data provider knows a priori the background knowledge of possible attackers and how the data will be used. These models use these assumptions to specify data types as "quasi-identifiable" or "sensitive". However, it is not so simple to make these assumptions about social networks. It is not easy to predict how applications may use social network data nor can concrete assumptions be made about the background knowledge of those who may attack a social network user's privacy. As such, all social network data must be treated as both sensitive and quasi-identifiable which breaks the existing anonymity models.

This paper discusses how the interactive data release model, used by social network APIs, may be utilized to provide anonymity guarantees without bounding attacker background knowledge or knowing how the data might be used. It is demonstrated that this data release model and anonymity definition provide applications with greater utility and stronger anonymity guarantees than would be provided by anonymizing the same data set using traditional methods and releasing it publicly.

This new anonymity model measures the maximum posterior probability that some user is associated with their private data conditioned on the data released by a social network API query. We call this anonymity model PP-anonymity (**P**osterior-**P**robability), and evaluate the feasibility of providing such a guarantee with a social network API.

II. Related Work

Privacy within the context of social networks is becoming a very hot topic in both research and among the public. This is largely due to an increase in use of Facebook and a set of high-profile incidents such as the de-anonymization of public data sets [3]. However, public concern about privacy has not necessarily translated into more responsible usage of the existing privacy mechanisms. It is suggested that this may be due to the complexity of translating real-world privacy concerns into online privacy policies, as such it has been suggested that machine learning techniques could automatically generate privacy policies for users [4]. Research into anonymizing data sets (or microdata releases) to protect privacy directly apply to the work in this paper. Most of this research has taken place within the database research community. In 2001, Sweeney published a paper [8] describing the "re-identification" attack in which multiple public data sources may be combined to compromise the privacy of an individual. The paper proposes an anonymity definition called k-anonymity. This definition was further developed and new approaches to anonymity were proposed that solved problems with the previous approaches. These later anonymity definitions include p-sensitivity [9], ℓ-diversity [10], t-closeness [11], differential privacy [12], and multi-dimensional k-anonymity [13]. All of these privacy approaches and their associated terms are discussed in section III.

Methods have been developed that attempt to efficiently achieve anonymization of data sets under certain anonymity definitions. Initially simple methods such as suppression (removing data) and generalization have been used to anonymize data sets. Research has sought to optimize these methods using techniques such as minimizing information loss while maximizing anonymity [14]. One approach called "Incognito" considers all possible generalizations of data throughout the entire database and chooses the optimal generalization [15]. Another approach called "Injector" uses data mining to model background knowledge of a possible attacker [16] and then optimizes the anonymization based on this background knowledge.

It has been shown that checking "perfect privacy" (zero information disclosure), which applies to measuring differential privacy and arguably should apply to most anonymity definitions, is Π_2^p-Complete [17]. However, other work has shown that optimal k-anonymity can be approximated in reasonable time for large datasets to within $O(k \log k)$ when k is constant, though runtime for such algorithms is exponential in k. It has been shown that Facebook data can be modeled as a Boolean expression which when simplified measures its k-anonymity [18]. Section VIII discusses how such expressions, constructed from Facebook data, are of a type that can be simplified in linear time by certain logic minimization algorithms[19].

Privacy in social networks becomes more complicated when the social applications integrate with mobile, sensing, wear-

able, and generally context-aware information. Many new mobile social networking applications in industry and research require the sharing of location or "presence". For example, projects such as Serendipity [1] integrate social network information with location-aware mobile applications. New mobile social applications such as Foursquare, Gowalla, etc... also integrate mobile and social information. Research suggests that this trend will continue toward seamlessly integrating personal information from diverse Internet sources, including social networks, mobile and environmental sensors, location and historical behavior [2]. In such mobile social networks, researchers have begun to explore how location or presence information may be exchanged without compromising an application user's privacy [5], [6]. Furthermore, other mobile information such as sensors may be used to drive mobile applications and this brings up issues of data verification and trust that are discussed here [7]. Thus, the ideas introduced in this paper will likely expand in applicability as social networks are extended into the mobile space.

III. Defining Anonymity

This section discusses the basic terms used in this paper. It breaks them up into data types, anonymity definitions, and anonymization techniques.

A. Data Types

The data sets most often considered in anonymization research usually take the form of a table with at least three columns that usually include zip code, age, (sometimes gender), and a health condition. This data set is convenient for many reasons, including its simplicity, but also contains (and doesn't contain) representative data types that are important to anonymization. First, it does not contain any ***unique identifiers*** such as Social Security numbers. The first step in anonymizing a data set is removing the unique identifiers. The most common unique identifiers discussed in this paper are social network user IDs (Facebook ID or username). The data set also contains a set of ***quasi-identifiers*** - age, zip code, and gender are the most common. Data may be considered a quasi-identifier if it can be matched with other data (external to the data set) which maps to some unique identifier. The re-identification attack consists of matching a set of quasi-identifiers from an anonymized data set to a public data set (such as census or voting records) effectively de-anonymizing the anonymous data set. It is important to note that quasi-identifiers are assumed to be public (or possibly public) by definition and as such are not the primary data to be protected. The data that are to be protected from re-identification are termed ***sensitive attributes***. Sensitive attributes are not assumed to be publicly associated with a unique identifier and as such their relationship to the quasi-identifiers within a data set is what concerns most anonymity definitions. In most research examples health conditions or disease attributes are considered sensitive attributes. A set of sensitive attributes that share the same set of quasi-identifiers are, together with their quasi-identifier set, called an ***equivalence class***. For example, the health conditions associated with 25 year old males living in a particular zip code would be an equivalence class.

Furthermore, the $\{$*zip code, gender, age, disease*$\}$ data set is useful because its data exhibit different characteristics. Zip codes are structured hierarchically and ages are naturally ordered. The rightmost digits in zip codes can be removed for generalization and ages can be grouped into ranges. Gender presents a binary attribute which cannot be generalized because doing so would render it meaningless. Finally, using disease as the sensitive value is convenient since health records are generally considered to be private. Also, disease is usually represented as a text string which presents semantic challenges to anonymization such as understanding the relationship between different diseases.

B. Anonymity Definitions

K-Anonymity [8] states that a data set is k-anonymous if every equivalence class is of size k (includes at least k records). However, it was observed that if the sensitive attribute was the same for all records in an equivalence class then the size of the equivalence class did not provide anonymity since mapping a unique identifier to the equivalence class was sufficient to also map it to the sensitive attribute; this is called ***attribute disclosure***.

p-sensitivity [9] was suggested to defend against attribute disclosure while complementing k-anonymity. It states that along with k-anonymity there must also be at least p different values for each sensitive attribute within a given equivalence class. In this case, an attacker that mapped a unique identifier to an equivalence class would have at least p different values from which only one correctly applied to the unique identifier. One weakness of p-sensitivity is that the size and diversity of the anonymized data set is limited to the diversity of values in the sensitive attribute. If the values of the sensitive attribute are not uniformly distributed across the equivalence classes there will be significant data loss even for small p values.

ℓ-diversity [10] was suggested to prevent attribute disclosure through either requiring a minimum of "entropy" in the values of the sensitive attribute or by placing a minimum and maximum on how often a particular value may occur within an equivalence class. While preventing direct attribute disclosure such an anonymization may result in the distribution of sensitive attribute values being significantly skewed. If the distribution of a sensitive attribute is known, this knowledge could be used to calculate the probability of a particular sensitive attribute value being associated with a unique identifier. For instance, while only 5/1000 records in a data set contain a particular disease, there may exist an equivalence class in the anonymized data set for which half the records contain the disease, implying that members of the equivalence class are 20 times more likely to have the disease.

t-closeness [11] approaches the problem of skewness by bounding the distance between the distribution of sensitive attribute values in the entire data set and their distribution within each equivalence class. The problem (or trade-off) with t-closeness is that it achieves anonymity by limiting the

statistical difference between equivalence classes and in doing so minimizes any interesting correlations or statistics that could be drawn from the anonymized data set. Furthermore, it is not clear that there is any efficient way to enforce t-closeness on a large data set [20].

Defending against skewness attacks presents a paradox - data sets are useful because they contain correlations that say something about the world outside of the data set, which is what a skewness attack does. In this sense the utility of a data set and its danger to privacy are correlated. Skewness attacks should therefore be approached practically considering the nature of the sensitive attributes in terms of the danger of their compromise and the utility they provide by being released.

Multi-Dimensional K-Anonymity [13] proposes a more flexible approach to K-anonymity in which equivalence classes are clustered or generalized across a table in more than one dimension. This flexibility allows for a higher degree of optimization than simply generalizing each column of a database separately. While optimizing the selection of equivalence classes is NP-hard, a greedy approximation algorithm for multi-dimensional K-anonymity has been shown to outperform exhaustive optimal algorithms for a single dimension.

Differential Privacy [12] takes a different perspective on privacy than the other privacy models discussed in this paper. Most interestingly, it assumes an interactive database model in which, as opposed to a non-interactive microdata release, the data collector provides an interface through which users may query and receive answers. As will be discussed in section IV, this model fits that currently used by many social network APIs and is much more practical for the types of data use associated with social networks. However, differential privacy focuses primarily on statistical databases, the queries on which are answered with added noise which guarantees a maximum level of danger to the privacy of anyone participating in the database. The difficulty in applying this to social networks is in appropriately measuring or defining "noise" in a way that meaningfully integrates with the data's use by social network applications. While interesting, this paper does not deal with the same problem. However, the interactive database model assumed by differential privacy is promoted as the appropriate model for anonymity mechanisms applied to social networks.

C. Anonymization Techniques

Finally, anonymization commonly consists of generalizing, perturbing, or suppressing data. Generalization of data requires some ordering or structure to the data type such that many specific values of data can be grouped as a related, but more general value. Perturbation involves distorting or adding noise to a value. Some types of data such as images may be perturbed and still useful. However, much social network data may not be useful when modified or generalized and as such must be removed or suppressed - as such suppression is the most generally applicable approach to anonymization when one cannot make assumptions about how generalization or perturbation will affect the utility of the data. Also, it should be noted that in social networks it is very common for a data field to have many values separated by commas. When items are suppressed from such data fields it could be said that the value of the data field has been generalized - however, this paper will refer to such an anonymization technique as suppression.

IV. Social Network Data

This section will highlight the difficulties of applying existing anonymity definitions and models to social network data. Most anonymity research assumes the same convenient data set discussed in section III. This data set contains a few quasi-identifiers that are usually hierarchical or ordered such that they can be easily generalized along with a clearly identifiable sensitive attribute. Furthermore, anonymity research has generally assumed a rather research-centric non-interactive data release model.

A. Data Characterization

The traditional anonymity data set consists of some version of $\{$*zip code, gender, age, disease*$\}$. This data set is easy to understand and naturally translates to privacy examples since it uses traditionally accepted quasi-identifiers and sensitive data. Social networks do not provide such convenient data.

Social network data often consists of attributes such as name, unique ID, friendship links, favorite movies/music/activities, birthdate, gender, hometown, group associations, guestbook or wall posts, pictures, videos, messages, status updates, and sometimes current location. To simplify discussion of social network data in this section we will assume the usage model and data types of the largest social network, Facebook.

The first task in anonymizing social network data is to specify which attributes are unique identifiers, quasi-identifiers and sensitive attributes. Obviously, the unique ID is a unique identifier and some people may wish that their name be considered a unique identifier as well. There are then the traditional quasi-identifiers including city, birthdate, and gender - however these data types are often targeted by Facebook applications as the attributes of interest (such as birthday calendar applications), and may be considered sensitive attributes by some users. In fact, depending on a user's privacy settings nearly every data attribute may be publicly available or semi-public within a region or network. Furthermore, these privacy settings are constantly changing and the user's privacy expectations may change drastically depending on context. Given the lack of clear assumptions as to the public availability of most information on Facebook, all data types should be considered a quasi-identifier. Also, given the complex nature of social network applications, (e.g., calendars of friends' birthdays, context-aware music players, or location-aware games) all attributes may potentially be considered sensitive attributes within certain contexts and as such all data types should be considered sensitive attributes.

This poses significant problems to utilizing traditional anonymity solutions for social networks. If a single attribute is considered both quasi-identifiable and sensitive it renders

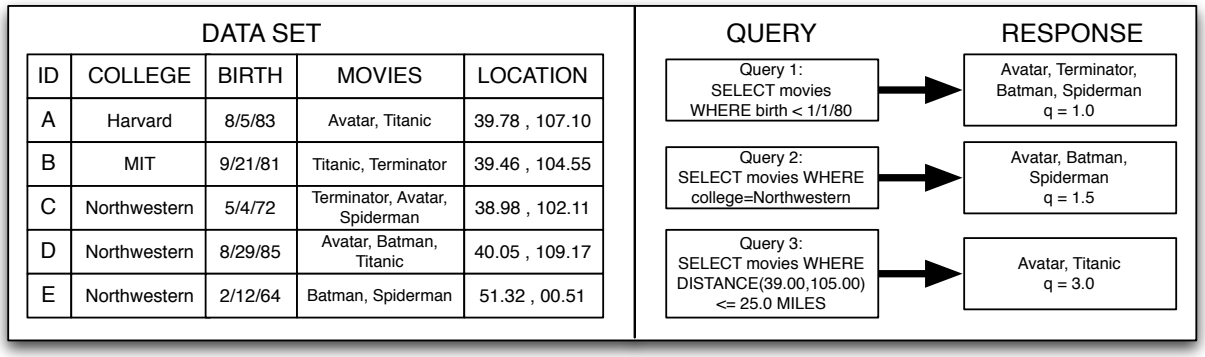

Fig. 1. Example data set, queries, and responses with associated q values for different PP-anonymity

k-anonymity incompatible with ℓ-diversity, p-sensitivity, and t-closeness. This is because equivalence classes must share the same quasi-identifier set (have the same values for all quasi-identifier attributes) and ℓ-diversity, p-sensitivity, and t-closeness require some variation of all sensitive attributes. t-closeness, ℓ-diversity, and p-sensitivity all assume equivalence classes defined by a shared quasi-identifier set. If some attribute was both sensitive and quasi-identifiable then it would be required to have the same value throughout the equivalence class (to avoid re-identification) and it would be required to have different values (to avoid attribute disclosure) rendering ℓ-diversity, p-sensitivity, and t-closeness [1] meaningless.

B. Data Usage

The obstacles to applying traditional anonymity approaches to social networks arise largely from the assumption that data is released in a non-interactive form as a database or table. Such a table results in each set of quasi-identifiers being uniquely associated with its sensitive attributes. However, this type of data association may not be necessary for many applications. Furthermore, a particular application may only be interested in a particular set of equivalence classes which include only a subset of quasi-identifiers. However, an alternative to the non-interactive data release model exists. In an interactive model, a "trusted" data collector provides an interface through which users or applications may query for information through an API or general language like SQL. Fortunately, Facebook already offers both a function API and a SQL-like language called FQL. Facebook's interface is currently used by over 500,000 applications and most major websites on the internet. Recent developments, such as the announcement that Facebook will soon integrate users' location information and automatically release this information to "trusted" third-parties without the user's explicit approval, provide a clear motivation for solutions to anonymity within an interactive data release model.

The next section defines the re-identification or anonymity problem assuming an interactive model and assuming that all data attributes are both quasi-identifiable and sensitive.

[1]being quasi-identifiable, the attribute must have a uniform distribution of one value

V. The Anonymity Problem in Social Networks

This section proposes a new anonymity problem in social networks that more closely conforms with the data types characteristic of social networks as well as the interactive data release model supported by social networks. In this new anonymity problem, a trusted data collector wishes to provide an interface through which applications may query user data without compromising the user's privacy beyond some threshold. Generally, a query is considered admissible if its released information does not result in any previously unknown mapping between data and identity having an implied posterior possibility greater than some threshold. We will refer to anonymous interactive interfaces that provide guarantees on such posterior probability as being "PP-anonymous". This section will also define how the interactive interface functions and the assumed background knowledge of the adversary.

PP-Anonymous Definition: We use the definition of "posterior probability" as the probability that an uncertain proposition is conditionally known by some adversary after the relevant evidence from an event is taken into account. In this case, the event is a re-identification attack in which some previously unknown mapping between a piece of private data and an identity is known to exist with some posterior probability conditional to the data released by a query. Therefore, given some value q - a query is PP-anonymous if the posterior probability of the existence of a previously unknown mapping between identity and data is greater than $\frac{1}{q}$.

Adversary Definition: Furthermore, we assume that the adversary may have access to any entry in the data set. Therefore, in spite of its general impossibility for statistical databases [12] we will accept the intention of Dalenius' desideratum for statistical databases: "that nothing about an individual should be learnable from the database that cannot be learned without access to the database." We define learned in terms of posterior probability of existence.

Interactive Interface Definition: For the purposes of this paper, we define an interactive interface, using traditional anonymity terms, as an interface that allows applications to specify an equivalence class through a query along with attributes of interest. The interface returns the values of the attributes as one field or array not specifying any mapping between table entries and attribute values. The interface may return zero or more values from each entry in the equivalence

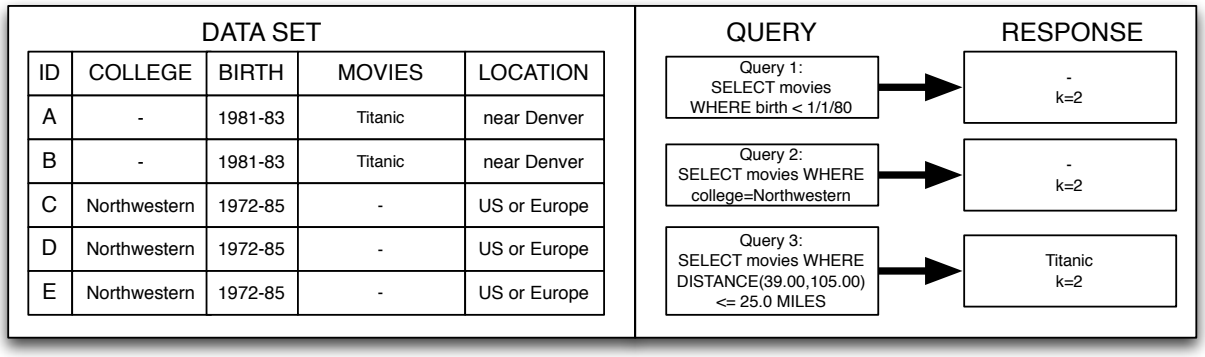

Fig. 2. Example data set K-anonymized with $k = 2$

class.

Privacy Compromise Definition: A privacy compromise has occurred if PP-anonymity, with a given q value, is violated by a response to some query.

It should be noted that if a query specifies an equivalence class that has fewer than q entries, the adversary may know that at least some of the released data is mapped to fewer than q identities, hence PP-anonymity implicitly requires all equivalence classes specified by a query to be at least size q. This requirement relates to the original K-anonymity definition.

Since the query response from the interactive interface does not include mappings between table entries and attribute values, the set of values may map to many different groups of individuals, each of which may fully account for all values in the query response. The next section discusses how the existence of these different groups, all of which account for a data release, may be measured to infer the probability that a set of private data is mapped to a specific individual.

VI. Measuring Anonymity

This section will discuss a practical approach to calculating bounds on knowledge learned by an adversary that may know every data value in a data set except one (the one we are theoretically calculating the bound for). By "bounding knowledge" we mean not releasing information that implies a posterior probability of association between an an individual and their private data greater than $\frac{1}{q}$. In calculating this bound for a particular query response we say we are *measuring the privacy of the query response*. Hence, the larger the q value the greater the PP-anonymity.

This section will explain ways to measure q, starting from the perspective of the adversary. A possible attack will be described which will imply a way that q can be calculated. In this way, the value of q will be concretely connected with its privacy implications from the beginning. Each example will be connected with the example data set, queries, and responses presented in Figure 1.

Rare Value Attack: Many strings found in social network data fields are very unique. For example, sometimes a user who likes the movie Avatar may include the string "Avatar totally rocks!!!" in their movie list, rather than simply including the string "Avatar". This string may uniquely map to the user and obviously should never be released from a social network's anonymous interface. Assume the string does uniquely map to this user and also say an attacker knows this and wants to find the location of the user. The attacker also knows that the anonymous interface only returns queries that are admissible under PP-anonymity with $q = 6$. The attacker could create 5 fake users at a particular location and list "Avatar totally rocks!!!" in their movie list. Then the attacker could query for the movie lists of users within a geographic area including the location of his five users. If the query is responded to with the string "Avatar totally rocks!!!" the attacker would know that the victim user was also in that geographic area, since there must have been at least 6 users in that area with the unique movie string or else the query would not have included it in the response.

This presents a rather simple way to measure privacy. The anonymous interface could simply measure the privacy of a query response by taking the least common value from the query response and counting the number of individuals in the equivalence class[2] that are associated with that value. We will refer to this value as the Least Common Value count (LCV count). As can be seen in Query 1 in Figure 1, the response contains four movies, one of which (Batman) is only listed by one individual in the specified equivalence class (those born before 1980). Hence this response appropriately lists $q = 1$. However, the next example will show that this measure of privacy is only an upper bound on privacy knowledge and misrepresents what the attacker may logically deduce from a query response.

Logical Exclusion Attack: Consider the response to Query 2 in Figure 1. Each of the three movies in the response has at least two individuals in whose movie lists they are included and might be considered PP-anonymous with $q = 2$. However this is not necesarilly the correct bound on posterior probability of associating the set of movies with an individual in the data set. The attacker could logically deduce that one of three possibilities must exist, the group correctly accounting for the data must contain either persons C and D, D and E, or C and E. These three groups represent the minimal combinations of which at least one must exist within any group which fully accounts for the data released by the query. This could be expressed as the boolean expression $(CD+DE+CE)$. Given that each individual is in two of the three groups the attacker can assume that there is a $\frac{2}{3}$ (or 66%) possibility that one of

[2]remember the equivalence class is specified by the query paramaters

those individuals is correctly associated with the data, hence $q = 1.5$.

This presents a stronger measure of privacy but is more complicated to measure. To measure privacy in this way requires that one find the minimal set of groups required to fully account for the data being released. Section VII discusses how this can be done by representing the data values and their mapping to users as a boolean expression and finding the prime implicants using logic minimization. Representing social network data in this manner was originally suggested and discussed for use in a mobile social network application [18].

For comparison, the example of the data set is also presented in Figure 2 anonymized with traditional k-anonymity ($k = 2$). Note that optimal k-anonymization would depend on knowing the application intentions and attributes of interest beforehand. While this is not possible, the example data set is k-anonymized to retain as much useful data as possible for comparison. Furthermore, it should be noted that this data set cannot contain diversity (as discussed in section IV) and as such is trivially vulnerable to an attribute disclosure attack.

VII. Real-Time Anonymization

This section discusses the practical steps involved in measuring the q value for PP-anonymity. The steps generally consist of **(1)** Building a logical Sum-of-Products (SOP) expression from the data to be released by a query. **(2)** Minimizing the SOP expression to find the essential prime implicants. **(3)** Calculating q which is the number of occurrences of the most common literal (variable) divided by the number of terms (clauses) in the minimized expression. Each of these steps will be explained and then the section will finish with a discussion of how this approach would perform in the real-world.

Remember that a query specifies an equivalence class (e.g., "men under 25 in Denver") and an attribute of interest within that class (e.g., "Movies"). The query response consists of a set of values from the attribute of interest within the equivalence class. It is this set of values in the query response and its relationship to the equivalence class that is measured to calculate q.

A. Step 1: Building an SOP Expression

This section describes how a product-of-sums (POS) expression is created from an equivalence class specifying a set of users and their data. The POS expression can then be converted to a Sum-of-Products (SOP) expression by De Morgan's law or more efficient method.

Given an equivalence class E consisting of n user entries and m attributes for which S is the set of all attribute values. We define a set $E_i \in E$ as the set of users which map to a certain value $S_i \in S$. A POS expression is formed from E_i whereby every set $E_i \in E$ is mapped into a sum term (clause) in which every user $e \in E_i$ is a literal (variable). This POS expression is then converted to an SOP expression for minimization or minimized directly, understanding that the output must be a two-level SOP expression.

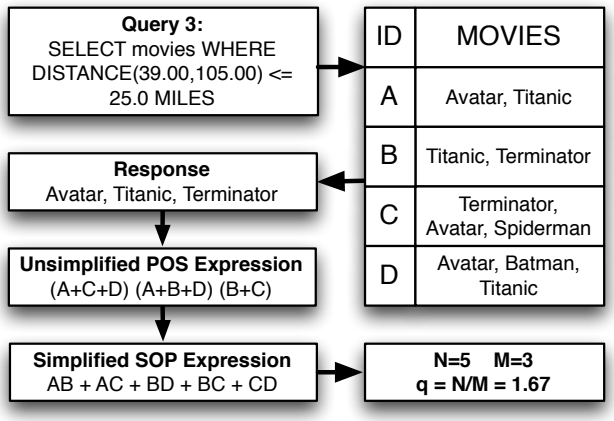

Fig. 3. Example of using logic minimization to calculate q

B. Step 2: Minimizing the Expression

The SOP expression is then minimized using a logic minimization method which finds the essential prime implicants of the expression. A classic algorithm for doing this is the Quine-McCluskey algorithm, developed in 1950's. This algorithm was used for initial evaluation of a related problem [18] and those results are discussed in section **??** along with discussion of how more recent developments in logic minimization may allow this step to scale linearly with the number of individuals and data items.

C. Step 3: Calculating q

Given a minimized SOP expression with n product terms the maximum number of times that any literal occurs (m) is divided by n. Hence $q = \frac{n}{m}$. Because the SOP expression is expressed partially with only the on-set - all product terms contain at most one occurrence of any literal. Therefore $n \geq m$ meaning that $q \geq 1$ and since all literals must occur at least once $q \leq n$.

D. Example

To help explain the process of measuring PP-anonymity we will reconsider Query 3 from Figure 1, however this time q will be calculated for a different query response. This example will be explained using Figure 3. The query requests the "MOVIE" values for those individuals within a geographic area specified in latitude and longitude. This area includes individuals {A,B,C,D}. In this case, q is being calculated for the response {Avatar, Titanic, Terminator}. As can be seen in Figure 3: Avatar is mapped to {A,C,D}, Titanic to {A,B,D} and Terminator to {B,C}. This results in 9 possible groups which could possibly account for the entire response set, these groups are represented by the unsimplified expression in Figure 3. Minimizing the unsimplified expression results in an expression with 5 product terms. Since there are 5 product terms in the unsimplified expression and no literal occurs more than 3 times $n = 5$ and $m = 3$, hence $\frac{n}{m} = q = 1.667$.

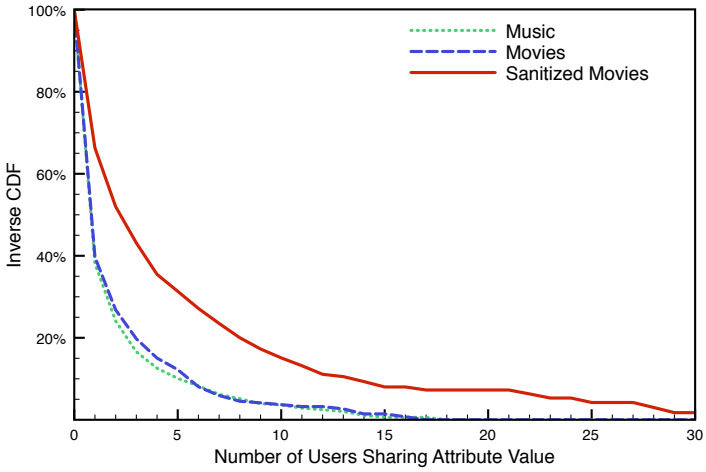

Fig. 4. Distribution of Attribute Values.

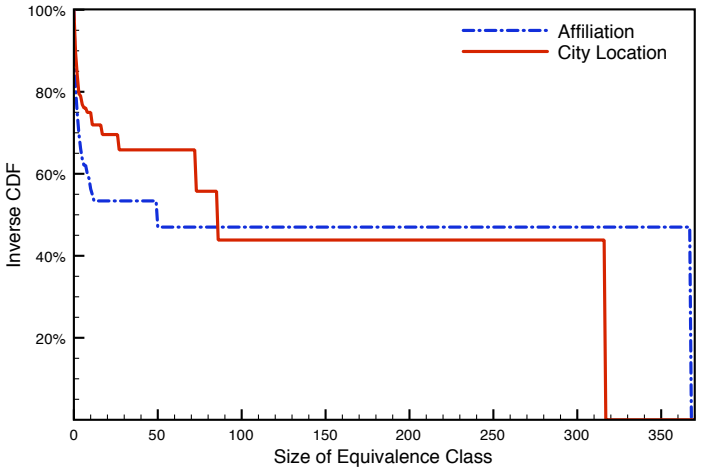

Fig. 5. Equivalence Class Size Distribution.

VIII. Evaluation

This section evaluates a set of Facebook data including over 700 users' city location, network affiliation, music, and movie preferences. The users in data set are all friends with the same user and could be considered an equivalence class specified by this friendship. As such, the data samples are not random but may be considered realistic for socio-digital systems in which social connections are natural aggregators. The distributions of these data are presented to understand the practical size of some equivalence classes and the distribution of their attribute data. These values are discussed in terms of q and what values of q are actually practical for a social network such as Facebook. Finally, a basic prototype is presented to show the feasibility of measuring q for social network data applications with discussion of how advanced logic minimization algorithms might scale in relation to social networks the size of Facebook.

A. Size of Equivalence Classes

In an interactive interface, equivalence classes may be specified with conditionals using sql style interfaces or particular API calls. Examples evaluated in this section include selecting those users in a particular city or users affiliated with a particular network. The size of the equivalence class is obviously a bound on what q values may be calculated for its data. The distribution of city locations and network affiliations in the test data is shown in Figure 5. A few locations and affiliations dominate the distribution with nearly half of the users sharing either a location or network affiliation, a few other values account for 5-10% of all values. Within a social application such a distribution should not be surprising considering that proximity and affiliation are the basis for many social connections. However, what can be noted is that about 20-40% of all values considered specify equivalence classes containing about 1% of the overall users. These smaller equivalence classes consist of only 5-10 users and may not be able to support PP-anonymous queries with q values greater than 3 or 4. As such, a PP-anonymous interface with $q \geq 10$ would not release information to queries specifying these smaller equivalence classes.

B. Distribution of Attribute Values

However, a large equivalence does not itself guarantee any anonymity as it may not contain significant overlap in the attributes for which the query is interested. For instance, the music and movie preferences listed by the users in the test data contain significant diversity as seen in Figure 4. While many of the users may indeed list the same movies and music artists among their favorite they often will not use matching strings to refer to same artist or movie. For instance users may list "Lord of the Rings", "LOTR", "The Rings Trilogy", or even "I heart LOTR". This reduces the usefulness of the data and disallows release of over 95% of music and movie data values under PP-anonymity with $q \geq 10$. Facebook is currently moving toward standardizing these values (probably to increase their worth to Facebook) through a new feature called connections in which the user "likes" something (e.g., thumbs up, thumbs down) resulting in a canonical string for each artist or movie. To understand how this data might be distributed the movie values from the data set were partially sanitized by using google.com to search imdb.com for the un-sanitized string value and then replacing the value with the movie title of the first result. This approach found the correct movie for over 95% of the movie values. When sanitized, over four times as many movie values were able to be released under PP-anonymity with a $q = 10$ and the same amount of data released from the un-sanitized movie values with $q = 10$ could now be released with $q = 27$, significantly increasing anonymity and the utility of the data. Considering that these values of q were calculated over a small subset of Facebook data (less than .001% of Facebook users) it is possible that much larger values of q could be calculated considering only 1% of Facebook users. Furthermore, since q generally grows with the number of users being evaluated, the number of users which need to be evaluated could be bounded given a target q values.

C. Scalability

Even if the number of users to be evaluated is bounded, would it be practical to calculate q over, say, a million users for every API call? A previous work which considered the

feasibility of using logic minimization on social network data in the manner considered in this paper found that using Quine-McCluskey, an algorithm from the 1950's, the simple two level boolean expressions created from social network users and their data can be minimized in milliseconds for hundreds of users [18]. Furthermore, such minimization generally scales linearly for the Facebook data that was tested in that paper. The authors of this paper are currently working to use more advanced minimization techniques designed to scale linearly for simple two-level minimization problems like that involved in calculating q.

D. Current Work

A minimization method developed at Czech Technical University has been shown to scale linearly for two-level logic minimization problems in which there are many terms (millions) and far fewer literals[19]. This algorithm is being evaluated for use in calculating q due to the fact that when there are n users (literals) and m attribute values, the number of terms to be minimzed is bounded by $O(n^{\frac{n}{4}})$ when $m \geq \sqrt{n}$, otherwise when $m < \sqrt{n}$ the number of terms is bounded by $O((\frac{n}{m})^m)$. While the number of terms has never been observed to scale anywhere near the maximum bound, it has been observed that as q increases so too does the ratio between terms to be simplified and the number of users (literals) in those terms. Whether or not q can be calculated for millions of users is still not known due to lack of access to a sufficiently large legitimate data set. However, q can be calculated over hundreds of users producing $q \leq 50$.

IX. Conclusion & Summary

The question of what constitutes weak or strong anonymity is still unanswered. To a large extent appropriate values for q will depend on the particular context and how the data being released relates to that context. In this sense, this paper does not claim that PP-anonymity can measure whether or not something is sufficiently anonymous. However, if practical values of q can be identified then those values could be used as the basis for policies governing use of social network APIs. For instance, API sessions could limit the number of queries within a certain amount of time or the API registration process could use q to limit or specify how, when and who will use the data. In short, having some way to measure the anonymity of a social network's API provides a tool with which to quantify and incorporate anonymity into social network data policies and practices.

This paper has shown that existing anonymity measures cannot be applied to social network APIs, the most common form of private data release on the internet. An alternative anonymity definition was proposed (PP-anonymity) which assumes an interactive interface model like that used by social network APIs. PP-anonymity measures the posterior probability that an all-knowing attacker may use an API query response to deduce which data maps to which user. Finally, real Facebook data was analyzed to understand what levels are PP-anonymity are practical for social network applications.

References

[1] N. Eagle and A. Pentland, "Social serendipity: Mobilizing social software," *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 28–34, 2005.

[2] A. Beach, M. Gartrell, X. Xing, R. Han, Q. Lv, S. Mishra, and K. Seada, "Fusing mobile, sensor, and social data to fully enable context-aware computing," in *HotMobile '10: Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*. New York, NY, USA: ACM, 2010, pp. 60–65.

[3] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 111–125.

[4] L. Fang and K. LeFevre, "Privacy wizards for social networking sites," in *WWW '10: Proceedings of the 19th International World Wide Web Conference*, 2010.

[5] K. P. N. Puttaswamy and B. Y. Zhao, "Preserving privacy in location-based mobile social applications," in *HotMobile '10: Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*. New York, NY, USA: ACM, 2010, pp. 1–6.

[6] L. P. Cox, A. Dalton, and V. Marupadi, "Smokescreen: flexible privacy controls for presence-sharing," in *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*. New York, NY, USA: ACM, 2007, pp. 233–245.

[7] P. Gilbert, L. P. Cox, J. Jung, and D. Wetherall, "Toward trustworthy mobile sensing," in *HotMobile '10: Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*. New York, NY, USA: ACM, 2010, pp. 31–36.

[8] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, October 2002.

[9] T. Truta and B. Vinay, "Privacy protection: p-sensitive k-anonymity property," in *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, 2006, pp. 94–94.

[10] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[11] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proceedings of the IEEE ICDE 2007*, 2007.

[12] C. Dwork, "Differential privacy," *Automata, languages and programming*, pp. 1–12, 2006.

[13] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, 2006, pp. 25–25.

[14] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Trans. on Knowl. and Data Eng.*, vol. 13, no. 6, pp. 1010–1027, 2001.

[15] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: efficient full-domain k-anonymity," in *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2005, pp. 49–60.

[16] T. Li and N. Li, "Injector: Mining background knowledge for data anonymization," in *ICDE '08: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 446–455.

[17] A. Machanavajjhala and J. Gehrke, "On the efficiency of checking perfect privacy," in *PODS '06: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. New York, NY, USA: ACM, 2006, pp. 163–172.

[18] A. Beach, M. Gartrell, and R. Han, "Social-k: Real-time k-anonymity guarantees for social network applications," in *IEEE International Workshop on SECurity and SOCial Networking (SESOC) at PerCom 2010*, 2010.

[19] P. Fiser and D. Toman, "A Fast SOP Minimizer for Logic Funcions Described by Many Product Terms," in *Proceedings of the 2009 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools*. IEEE Computer Society, 2009, pp. 757–764.

[20] J. Domingo-Ferrer and V. Torra, "A critique of k-anonymity and some of its enhancements," in *ARES '08: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 990–993.