Privacy-Preserving Social Media Data Outsourcing

Jinxue Zhang*, Jingchao Sun*, Rui Zhang†, Yanchao Zhang*, and Xia Hu‡
*School of Electrical, Computer and Energy Engineering (ECEE), Arizona State University
†Computer and Information Sciences Department, University of Delaware
‡Computer Science & Engineering, Texas A&M University
*{jxzhang, jcsun, yczhang}@asu.edu, †ruizhang@udel.edu, ‡hu@cse.tamu.edu

***Abstract*—User-generated social media data are exploding and of high demand in public and private sectors. The disclosure of intact social media data exacerbates the threats to user privacy. In this paper, we first identify a text-based user-linkage attack on current data outsourcing practices, in which the real users in an anonymized dataset can be pinpointed based on the users' unprotected text data. Then we propose a framework for differentially privacy-preserving social media data outsourcing for the first time in literature. Within our framework, social media data service providers can outsource perturbed datasets to provide users differential privacy while offering high data utility to social media data consumers. Our differential privacy mechanism is based on a novel notion of ϵ-text indistinguishability, which we propose to thwart the text-based user-linkage attack. Extensive experiments on real-world and synthetic datasets confirm that our framework can enable high-level differential privacy protection and also high data utility.**

I. Introduction

User-generated social media data are exploding. Twitter, the most popular microblogging service, generates 500 million tweets per day by 328 million monthly active users as of June 2017. Facebook, the largest online social network with about 1.3 billion daily active users as of June 2017, generates petabytes of data per day. People use social media platforms to communicate with their friends, share their daily life experiences, express their opinions on political/social events and commercial products, etc.

Closely tied to human beings in the physical world, large-scale social media data have tremendous usages by various *data consumers* and have become one of the most profitable resources for social media service providers [1]. For example, companies use social media data to study customer behavior, monitor public responses to their products, deliver online advertisements more cost-effectively, and uncover the trends that may impact their businesses; public policy makers explore social media data to obtain the demographic information for making strategic decisions; and sociologists leverage social media data to study the social behavior and establish new social network theories. In a typical social media mining application, a data consumer demands a set of users and their social media data (such as profiles, recent posts, and friends), which satisfy some desirable criterion. For example, company A may request the data of all the users who mentioned the company name in the past week after a public relation crisis.

The disclosure of intact social media data exacerbates the threats to user privacy. For example, many users mention their vacation plans in publicly visible tweets without knowing that criminals can exploit such information for targeted break-ins and thefts [2]. Criminals may identify potential victims nearby by directly browsing/searching social media platforms, and smarter ones can explore the search APIs offered by social media platforms. The data acquired in this traditional way are only small and random samples of all the qualifying data. For example, Twitter claims that their Search API only "searches against a sampling of recent tweets published in the past 7 days" [3]. If the criminals could access intact social media data relevant to the target area, they can identify all potential victims to plan large-scale break-ins. In addition, social media mining applications are increasingly sophisticated and powerful. If intact social media data are available, lots of sensitive information the users do not explicitly disclose could still be inferred, such as age [4], [5], location [6], [7], language [8], and political preferences [9].

There is a natural conflict between data utility and user privacy in social media data outsourcing. On the one hand, data consumers want intact social media data to maximize the data utility, which is also the most profitable case for social media service providers. The maximum data utility is achieved unfortunately at the biggest sacrifice of user privacy. On the other hand, social media service providers are also motivated to protect user privacy due to legal concerns, public relations, and many other reasons. For example, they may intentionally add random noise to the data before releasing. User privacy is thus better protected but at the loss of data utility.

A growing body of work studies privacy-preserving outsourcing of social graphs and falls into two directions. The first line of research [10], [11] aims at vertex privacy by outsourcing social graphs with anonymous user IDs, and the effort is to prevent the adversary from linking anonymous IDs to corresponding users in the real social network. The other line of research targets link privacy [12]–[14], and the main effort is to outsource social graphs with real user IDs but perturbed links by deleting real edges and adding fake ones. Neither line of work considers the privacy of user data and thus cannot be directly applied in our context.

In this paper, we propose a framework for privacy-preserving social media data outsourcing. The framework consists of a data service provider (DSP), numerous social media users, and a lot of data consumers. The DSP can be either a social media service provider itself such as Twitter or Facebook, or a third-party data company such as Gnip and

DataSift which resells the data obtained from social media service providers. Data consumers can be an arbitrary individual or entity in public or private sectors. They are interested in statistical information that can be mined from social media data, rather than real user IDs. A data consumer submits a data request to the DSP, which specifies the query conditions. The DSP responds with social media data satisfying the query conditions, in which each user is anonymized.

Although there can be various attacks on social media data outsourcing, we consider a *user-linkage attack* as the first effort along this line. In this attack, a malicious data consumer attempts to link random or selected anonymous IDs in the received data set to real IDs on the social media platform, so he can obtain the latest social media data about the victims or other sensitive information not covered by his previous query. We assume that existing sophisticated techniques such as [10], [12]–[14] are adopted to preserve both link privacy and vertex privacy in the anonymized data set, so the attacker cannot uncover real IDs based on either vertexes or edges.

Our defense against the user-linkage attack in this paper consists of three steps. First, we map the intact data of all the users into a high-dimensional user-keyword matrix. Second, we add controlled noise to the user-keyword matrix to satisfy differential privacy [15], the most popular privacy model lately. Finally, the perturbed user-keyword matrix is disclosed to the data consumer, where each user ID is anonymized. If the social graph corresponding to the data set is also needed, existing defenses such as [10], [12]–[14] should be adopted to preserve both link privacy and vertex privacy. Our defense applies to a wide range of social media applications. For example, the data consumer can infer demographic information about the target population from the perturbed data set.

Our contributions can be summarized as follows.

- We are the first to coin the problem of privacy-preserving social media data outsourcing to the best of our knowledge, for which a system model is also proposed.
- We propose a novel mechanism to guarantee differential user privacy while maintaining high data utility in social media data outsourcing. The popular Laplacian mechanism to achieve differential privacy suffers from the *curse of dimensionality* [16] and can bring huge noise to the original dataset which significantly reduces data utility. We define a new metric called ϵ-text indistinguishability whereby to design a mechanism to break this constraint.
- We thoroughly evaluate the proposed defense on a real-world dataset with regard to user privacy and data utility. Our results show that high-level privacy protection can be achieved without significantly sacrificing data utility. For example, we show that our mechnism can reduce the privacy leakage by as much as 64.1% by reducing only 1.61% of utility in terms of classification accuracy.

II. Problem Statement

A. Social Media Data Outsourcing

We consider a system with three parties: social media users, social media data service providers, and data consumers.

Social media users use the social media to connect with their friends and/or ones they have followed and generate the original texts which could be set either *private* or *public*. Public users are searchable either directly via the social media service provider's website or APIs or from the external tool such as Google. By setting his/her profile private, a private user only allows the authenticated users to access the profile and is not searchable from other users. However, the social media service provider still has full access to all the private and public data per user agreements.

The social media data service provider (or DSP for short) hosts and provides most likely paid access to social media data. A DSP can be a social media service provider such as Twitter or Facebook itself. It can also be an emerging third-party data company such as Gnip or DataSift, which partners with social media service providers to provide social media data services. For example, Gnip and DataSift both have authorized access to Twitter's Firehost engine whereby to have access to complete, intact, and realtime Twitter data. The DSP can outsource the data according to the privacy policies and agreements which users consent to when signing up for using social media services. Generally, the DSP has full rights to use all the hosted data for their businesses and also share the data with data consumers. For example, the DSP can sample the whole user space according to data consumers' requests, assign an anonymous ID to each sampled user, process the original data from each user according to data requests, and finally deliver the processed data to data consumers.

Data consumers purchase social media data in the user-keyword format from the DSP whereby to run various social media mining algorithms for extracting useful information. Other types of social media data such as timestamps are out of this paper's scope. A data consumer can be an individual, a business, a government agency, a research institution, or any other entity in public and private sectors who is aware of the growing importance of social media data. A data consumer typically sends to the DSP a request specifying its query conditions, pays for the request, and then receives the user-keyword data. For example, company A may request the data of all the users in the west coast who have tweeted the keyword "company A" in the past week. After receiving the data from the DSP, it can explore advanced social media mining algorithms to identify critical market trends and analyze the users' demographic information such as age, location, education level, income level, marital status, occupation, religion, and family size. Data consumer currently cannot obtain intact social media data without the DSP's support.

B. Adversary Model (User-Linkage Attack)

The DSP is assumed to be fully trusted by both social media users and data consumers. Some advanced social media users may be privacy-aware and have taken some actions to protect their privacy. For example, the statistics in [17] and [18] show that 11.84% of Twitter users and 52.8% of Facebook users set their accounts private, respectively. As said, the DSP still has access to the complete data despite the users' privacy settings.

In addition, the users' effort to protect their privacy fails in the presence of the attack outlined below.

Our focus is to defend against the *user-linkage* attack, which can be launched by a curious or even malicious data consumer. Assume that the DSP has anonymized every user ID in the dataset and also taken existing defenses such as [10], [12]–[14] to guarantee link and vertex privacy. There are two possible versions of the user-linkage attack. In the first version, the attacker locates some target users by random browsing or searching via public APIs on the social media platform. It knows that these users must be in the received dataset under anonymous IDs. Existing defenses only consider link and vertex privacy via various obfuscation mechanisms, and no attention has been paid to text data. Armed with the text data of the target users with real IDs, the attacker can easily locate the corresponding anonymous IDs in the dataset. In the same way, the attacker can link the real IDs of the initial target users's friends to the corresponding anonymized IDs, and so on. The attacker eventually can uncover all the mappings between real and anonymous IDs in the dataset, despite the DSP's anonymization effort even based on existing advanced defenses [10], [11]. In the second version, the attacker tries to learn more beyond the received dataset. It starts by finding some interesting posts/tweets in the anonymized dataset and then easily locating the real users by performing simple text matching on the social networks. Once the real users are located, the attacker can learn their latest information.

C. Design Objectives

We consider the following problem within the aforementioned social media data outsourcing framework. After receiving a data query from the data consumer, the DSP searches the entire social media database to generate a dataset $\mathcal{D}$, which contains all the users satisfying the query and their outsourced texts (e.g., tweets, retweets, and replies) during the period specified in the query. Each user in $\mathcal{D}$ is assigned an anonymous ID to provide baseline user privacy. The data consumer may also request the social graph associated with $\mathcal{D}$, in which case we assume that existing defenses such as [10], [12]–[14] are adopted to preserve link and vertex privacy such that it is infeasible to link an anonymous ID to the real user based on his/her vertex's graphical property in the social graph. Our focus is to let the DSP transform the raw dataset $\mathcal{D}$ into a new one $\mathcal{D}'$ by perturbing the user texts according to the following three requirements.

- *Completeness*: each data item in $\mathcal{D}$ can be mapped to a unique item in $\mathcal{D}'$, and vice versa. In other words, no user is added to or deleted from $\mathcal{D}$ to create $\mathcal{D}'$.
- *Privacy Preservation*: The user texts in $\mathcal{D}'$ can be used to link any anonymous ID in $\mathcal{D}'$ to the real user with negligible probability, meaning that text-based user-linkage attacks can be thwarted with overwhelming probability.
- *High Utility*: $\mathcal{D}'$ and $\mathcal{D}$ should lead to comparable utility at the data consumer on common data mining tasks such as statistical aggregation, clustering, and classification.

III. Differentially Privacy-Preserving Social Media Data Outsourcing

In this section, we present a novel technique to achieve differentially privacy-preserving social media data outsourcing with the aforementioned design goals in mind. Inspired by geo-indistinguishability from [19], which is proposed to protect location privacy, we propose a novel notion of *text-indistinguishability* as the foundation of our technique.

A. Text Modeling

As stated before, social media service providers such as Facebook and Twitter currently outsource the original data set $\mathcal{D}$ to the data consumer, which contains the intact user texts. We assume that there are n users in $\mathcal{D}$, each assigned an anonymous ID. There are two obvious drawbacks here. First, although this method can enable the maximum data utility, it is vulnerable to the text-based user-linkage attack. Second, the data consumer cannot directly use the original texts which are highly unstructured and noisy, as mentioned in Section I. For example, common machine learning algorithms such as SVM and K-means require the input for each user to be a vector. Therefore, from the perspectives of both privacy protection and data usability, the DSP needs to transform each user's texts into a numerical vector. Here we introduce text modeling, a standard process to achieve it.

We first remove *stop words* in a stop-word list,[1] in which the words such as "the" and "those" are considered more general and meaningless. Then we conduct stemming [20] to reduce inflected words to their stem forms such that the words with different forms can be related to the same word. For example, "play", "playing", and "played" are all reduced to "play".

Next, we represent the keyword space for the cleansed texts using a τ-gram technique, which is widely used for statistical text analysis. The τ-gram technique splits a give message into sequences of τ contiguous words, each referred to as a τ-gram with τ ranging from 1 to the message length. For example, consider a tweet {"#SuperSunscreen is really useful, and I like its smell"}. After removing stop words and performing stemming, we have {"supersunscreen really useful like smell"}. The corresponding 1-grams are {"supersunscreen", "really", "useful", "like", "smell"}, and the corresponding 2-grams are {"supersunscreen really", "really useful", " useful like", "like smell"}. We let $\mathcal{N}_i$ denote the τ-grams of tweet corpus for each user $i \in [1, n]$ for all possible values of τ. Then we choose the top m most frequent τ-grams in $\bigcup_{1 \le i \le n} \mathcal{N}_i$, each of which is referred to as a *keyword* hereafter.

Finally, we use Term Frequency Inverse Document Frequency (TF-IDF) [21] to derive each element $D_{i,j}$ in the eventual dataset. Specifically, let $\Gamma(j)$ be the number of times a τ-gram j appears in the τ-gram list $\mathcal{N}_i$ of user i, $\Gamma_i^* = \max_{j \in \mathcal{N}_i} \Gamma(j)$, and $\Gamma'(j)$ be the number of users whose τ-gram lists contain j. We define

$$D_{i,j} = (0.5 + 0.5 * \frac{\Gamma(j)}{\Gamma_i^*}) * \log(\frac{n}{\Gamma'(j)}) \ . \tag{1}$$

[1] http://www.lextek.com/manuals/onix/

The above normalization is necessary because the users normally have very different tweet sets and thus different τ-gram lists. Interested readers are referred to [21] for more details about TF-IDF. We abuse the notation by letting $\mathcal{D} = [D_{i,j}] \in \mathbb{R}^{n\times m}$ denote the dataset after text modeling as well, which is essentially an $n \times m$ user-keyword matrix. We also let $U_i := \langle D_{i,1}, \ldots, D_{i,m}\rangle$ denote the text vector of user i ($i \in [1, n]$), i.e., the ith row in $\mathcal{D}$.

It is a common practice to use 1-grams and 2-grams only for high computational efficiency without significantly sacrificing the analysis accuracy. So the keyword space and user-keyword matrix can be constructed very quickly in practice. Also note that the DSP needs to outsource the $\tau-$gram name of each column. Otherwise, the data consumer has no idea about the physical meaning of the released data.

B. Why Differential Privacy?

The text model above has two important implications. First, it makes the unstructured social media data structured by reducing the keyword dimension from unlimited to m. Second, since the keyword space is composed of the top m most frequent τ-grams, the users' privacy has been largely improved in contrast to the original intact text data. For example, when a user has a tweet saying "The last class with students at CSE561, #MIT", the word "CSE561" or even "MIT" has very low probability to be selected in the keyword space. Therefore, this critical information has been hidden by the text modeling process. The privacy threat, however, cannot be completely eliminated. For instance, the 1-grams such as "last", "class", and "student" may still be released. These pieces of information can at least tell that the user is a professor or teacher. By combining other text information such as "computer" and "software," the attacker can further link the target user to a college professor teaching computer science. Such inferences can be continued until the target is linked to one or a few real IDs on the social media platform.

Differential privacy is a powerful technique to protect such linkage attacks. Proposed by Dwork *et al.* [15], differential privacy protects the individual user's privacy during the statistical query over a database. If each user in the database is independent, with any side information except the target him/herself, the attacker cannot infer whether the target user is in the database or which record is associated with him/her [22]. Providing arguably the strongest analytical protection for user privacy, the differential privacy model can be more formally defined as follows, which is tailored for our social media data outsourcing framework.

Definition 1 (ϵ-Differential Privacy [15])**.** *Given a query function $f(\mathcal{D})$ with an input dataset $\mathcal{D} \in \mathbb{R}^{n\times m}$ and a desirable output range, a mechanism $K(\cdot)$ with an output range $\mathcal{R}$ satisfies* ϵ-differential privacy *iff*

$$\frac{Pr[K(f(D_1)) = R \in \mathcal{R}]}{Pr[K(f(D_2)) = R \in \mathcal{R}]} \leq e^{\epsilon} \tag{2}$$

for any datasets $D_1, D_2 \in \mathbb{R}^{n\times m}$ that differ on only one row.

Here ϵ is the privacy budget. Large ϵ (e.g. 10) results in large e^{ϵ} and indicates that the DSP can tolerate large output difference and hence large privacy loss (because the adversary can infer the change of the database according to the large change of the query function $f(\cdot)$. By comparison, small ϵ (e.g., 0.1, $e^{0.1} = 1.1052$) indicates that the DSP can tolerate small privacy loss.

Differential privacy models can be interactive and non-interactive. Assume that the data consumer intends to execute a number of statistical queries on the same dataset. In the interactive model, the data consumer submits to the DSP the conditions for constructing the dataset $\mathcal{D}$ and also a desirable statistical query function f. Instead of returning $\mathcal{D}$ to the user, the DSP only responds with $K(f(\mathcal{D}))$, where $K(\cdot)$ perturbs the query result. In contrast, the DSP in the non-interactive model designs a mechanism $K(\cdot)$ to transform the original dataset $\mathcal{D}$ into a new dataset $\mathcal{D}' = K(f(\mathcal{D}))$. Finally, $\mathcal{D}'$ is returned to the data consumer which can execute arbitrary statistical queries locally.

C. ϵ-Text Indistinguishability: a New Notion

Our problem can be formulated according to a non-interactive differential privacy model as follows. Let us use an identity query $f_I(\cdot)$ as the query function such that $f(\mathcal{D}) = \mathcal{D}$. Our goal is to find a mechanism $K(\cdot)$ to transform the original user-keyword matrix (or dataset) $\mathcal{D}$ into a new one $\mathcal{D}' = K(\mathcal{D})$ such that ϵ-differential privacy can be achieved. Instead of transforming the entire dataset $\mathcal{D}$ as a whole, a more straightforward approach is to perform the transformation for each row individually, i.e., adding noise to each row $U_i \in \mathcal{D}$ to produce a new row $U_i' \in \mathcal{D}'$.

The Curse of Dimensionality. The Laplacian mechanism [15] is a popular technique for providing ϵ-differential privacy, but it suffers from the *curse of dimensionality*. To see it more clearly, recall that ϵ-differential privacy is defined over the query function f and unrelated to the dataset because Eq. (2) holds for all possible datasets. What matters is the maximum difference of $f(D_1)$ and $f(D_2)$ ($\forall D_1, D_2 \in \mathbb{R}^{n\times m}$), which is called the *sensitivity* of the query function f defined as

$$S(f) = \max \|f(D_1) - f(D_2)\|_1 \ . \tag{3}$$

As identity query $f_I(\cdot)$ transforms each text vector in $\mathcal{D}$ to a new vector in $\mathcal{D}'$, the sensitivity can be further defined as

$$S(f_I) = \max \|U_i - U_j\|_1 \tag{4}$$

where $U_i \in \mathbb{R}^m$ and $U_j \in \mathbb{R}^m$ are any two arbitrary vectors based on TF-IDF (see Eq. 1).

The Laplacian mechanism can achieve ϵ-differential privacy by adding the Laplacian noise to the query result [15], i.e.,

$$K_{Lp}(f_I(U_i)) = U_i + (Y_{i1}, \cdots, Y_{im}), i = 1, \ldots, n \ , \tag{5}$$

where Y_{ij} are drawn i.i.d. from $\text{Lap}(S(f_I)/\epsilon) \propto e^{-\epsilon|x|/S(f_I)}$.

The Laplacian mechanism unfortunately decreases the utility of the transformed dataset. Specifically, the larger the dimension m from the output of the identity query function $f_I(\cdot)$, the larger the sensitivity $S(f_I)$, the larger deviation of

the Laplacian noise. Moreover, the large noise accumulated from the high dimension will be added to each single element of $K_{Lp}(f_I(U))$, leading to the so-called curse of dimensionality. Specifically, from the definition of the text vector U_i in Eq. (1), the norm of each element in U_i should be less than $\log(n)(\approx 11.5$ when $n = 100000)$. When the dimension m (e.g., 10000) is large enough, the added Laplacian noise has deviation $O(m)$, which can easily exceed the norm of original text element (≈ 11.5).

ϵ-Text Indistinguishability. The root cause of the curse of dimensionality is that the noise added to a single element in every text vector U_i ($\forall i \in [1, n]$) is proportional with the L_1-sensitivity of U_i. To tackle this problem, we need to limit the sensitivity of the whole text vector to the norm of the vector, instead of the individual element.

To begin with, we need to generalize the concept of differential privacy defined in Definition 1. The generalization of differential privacy was first proposed by Andrés *et al.* for location privacy [19], where the privacy budget is proportional to the physical distance between any two users. They also propose the concept of geo-indistinguishability such that the service provider reports similar distribution with the difference bounded by $e^{\epsilon d(\mathrm{loc}_1, \mathrm{loc}_2)}$ for any two users at locations loc_1 and loc_2, respectively. Inspired by this work, we let $d(U_i, U_j)$ denote the Euclidean distance between U_i and U_j, which are any pair of text vectors in the user-keyword matrix $\mathcal{D}$. We further redefine the privacy budget as $\epsilon d(U_i, U_j)$ and propose the notion of ϵ-text indistinguishability.

Definition 2 (ϵ-Text Indistinguishability)**.** *Given the user-keyword matrix $\mathcal{D} = [D_{i,j}] \in \mathbb{R}^{n \times m}$, a mechanism $K_t(\cdot)$ satisfies* ϵ-Text Indistinguishability *iff*

$$\frac{Pr[K_t(U_i) = U^* \in \mathbb{R}^m | U_i]}{Pr[K_t(U_j) = U^* \in \mathbb{R}^m | U_j]} \leq e^{\epsilon d(U_i, U_j)}, \tag{6}$$

where U_i and U_j are any user text vector pair in $\mathcal{D}$, and U^* is a text vector in perturbed user-keyword matrix $\mathcal{D}'$.

The above definition means that any two vectors U_i and U_j in $\mathcal{D}$ can be transformed (or perturbed) by the mechanism $K_t(\cdot)$ into the same vector in $\mathcal{D}'$ with probability $\geq e^{-\epsilon d(U_i, U_j)}$. In other words, the more similar two text vectors are, the more non-distinguishable they are after transformation, and vice versa. The maximum privacy budget is given by $\epsilon r_{\max}$, where $r_{\max}$ denotes the maximum Euclidean distance between two text vectors in $\mathcal{D}$. As in the original ϵ-differential privacy mechanism, the larger the privacy budget, the larger the privacy loss the DSP can tolerate, and vice versa. Theorem 1 gives the upper bound of $\epsilon r_{\max}$, based on which the DSP can select ϵ to ensure an acceptable privacy budget.

Theorem 1. *Given the user-keyword matrix $\mathcal{D} \in \mathbb{R}^{n \times m}$ built according to Eq. (1), the maximum Euclidean distance between two text vectors is $r_{\max} \leq \sqrt{m}\log(n)$.*

Proof. According to the definition in Eq. (1), a text vector U has the maximum norm $\sqrt{m}\log(n)$ when each of its element is equal to the maximum value $\log(n)$. It follows that $r_{\max} \leq \|U_1 - U_2\| \leq \|U\| \leq \sqrt{m}\log(n)$. □

The upper bound above is almost unreachable in practice, as it requires that all the m keywords be used by only one user. So $r_{\max}$ is far less than $\sqrt{m}\log(n)$. But if the DSP chooses ϵ according to $\sqrt{m}\log(n)$, the effective privacy budget for many text-vector pairs is very small, implying that these text-vector pairs are very likely to be indistinguishable after perturbation.

D. Achieving ϵ-Text Indistinguishability

In this section, we propose a mechanism to achieve the ϵ-text indistinguishability. To this end, we first assume $r_{\max}$ to be infinite and then finite.

1) Mechanism for Infinite $r_{\max}$: The mechanism $K_t(f_I(\cdot))$, designed for the identity query $f_I(\cdot)$, maps each text vector $U \in \mathbb{R}^m$ of the dataset $\mathcal{D}$ to a new U' with the same dimension m. To that end, we write the perturbed U' as:

$$U' = U + d\Theta$$

where d is a random variable indicating the Euclidean distance between U and U', and Θ is an m-dimensional random vector drawn from the m-dimensional unit hypersphere. The mechanism is then composed of two steps: the generation of the magnitude and the direction. Since the drawing of Θ is straightforward, we focus on generating d. Similar to the Laplacian mechanism [15], we let d deviate from the center U by the Laplacian distribution,

$$g(d) = \epsilon e^{-\epsilon d} \tag{7}$$

where d ranges from zero to infinity. It is easy to check that $\int_0^{+\infty} g(d) = 1$.

The CDF of d is given by

$$C_\epsilon(d <= r) = \int_0^r \epsilon e^{-\epsilon x} dx = 1 - e^{-\epsilon r}. \tag{8}$$

The CDF above tells us how to generate a random d. Specifically, given a user text vector U, we want to generate a perturbed vector which has at most d Euclidean distance from U. Since d follows the C DF defined in Eq. (8), given a random probability $p \in [0, 1]$, we can obtain

$$d = C_\epsilon^{-1}(p) = -\frac{\log(1-p)}{\epsilon}. \tag{9}$$

We now show that the proposed mechanism satisfies ϵ-text indistinguishability.

Theorem 2. *The mechanism $K_t(f_I(\cdot))$ defined above achieves the ϵ-text indistinguishability.*

Proof. Given two user text vectors U_i and U_j, the probability quotient of being perturbed to the same vector U^* is

$$\begin{aligned} \frac{Pr[U = U^* | U_i]}{Pr[U = U^* | U_j]} &= \frac{Pr[d(U^*, U_i)]\Theta_1}{Pr[d(U^*, U_j)]\Theta_2} \\ &= e^{\epsilon(d(U^*, U_i) - d(U^*, U_j))} \leq e^{\epsilon(d(U_i, U_j))}. \end{aligned} \tag{10}$$

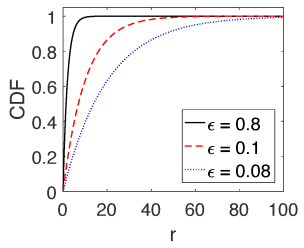


Fig. 1: The CDF of d with different ϵs.

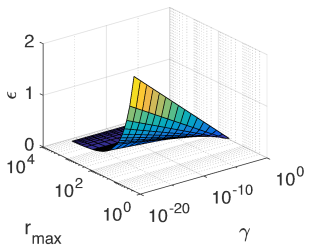


Fig. 2: Determine ϵ by γ and $r_{\max}$.

Here Θ_1 and Θ_2 can be canceled because both are drawn from the m-dimensional unit hypersphere with the same probability, and the inequity holds because of the triangle inequity. □

2) Mechanism for Limited $r_{\max}$: The mechanism $K_t(f_I(\cdot))$ in last section maps the user text vector U to U' with potentially infinite distance. However, we have demonstrated in Theorem 1 that any text vector pair have the Euclidean distance bounded by $r_{\max}$. Here we present how to truncate the mapping into a specific $r_{\max}$. We denote the corresponding mechanism as $K_r(f_I(\cdot))$.

As we can see from Fig. 1, $C_\epsilon(d = r)$ will approach to one quickly as r increases.

Therefore, we define a tolerance parameter γ to indicate how much of CDF will be outside $r_{\max}$. In other words,

$$1 - \gamma = C_\epsilon(d <= r_{\max}) = 1 - e^{-\epsilon r_{\max}} \Rightarrow \epsilon = -\frac{\log(\gamma)}{r_{\max}}. \tag{11}$$

The algorithm of $K_r(f_I(\cdot))$ is listed in Alg. 1. Given the tolerance parameter γ and $r_{\max}$, Line 1 computes the ϵ by Eq. 11. Then for each U_i in the dataset $\mathcal{D}$, we draw the noise vector $d\Theta$ by two steps: (1) obtain the magnitude in Line 3 by $r = C_\epsilon^{-1}(p)$, and (2) compute the direction Θ in Line 4 by drawing a random vector from the unit m-dimensional hypersphere. Line 5 adds the noise to get U'.

Algorithm 1: Perturbation alg. for mechanism $K_r(f_I(\cdot))$

Input : $r_{\max}, \gamma, \mathcal{D} = \{U_1, \ldots, U_n\}$
Output: Perturbed dataset $\mathcal{D}' = \{U'_1, \ldots, U'_n\}$
1 Compute ϵ according to Eq. (11);
2 For each $U_i, i = 1, \ldots, n$;
3 Select a random number $p \in [0, 1]$, and compute the radius d according to Eq. (9) ;
4 Select a random vector $N \in \mathbb{R}^m$, and normalized it to have unit L_2 norm, i.e., $\Theta = N/\|N\|_2$;
5 $U'_i = U_i + d\Theta$;

We also show that the mechanism $K_r(f_I(\cdot))$ achieves the ϵ-text indistinguishability.

Theorem 3. *The mechanism $K_r(f_I(\cdot))$ defined above achieves the ϵ-text indistinguishability within $r_{\max}$.*

Proof. For each text vector U, with the probability of $1 - \gamma$, the perturbed text vector U' has the Euclidean distance less or equal to $r_{\max}$ from U. The rest steps follow the proof of Theorem 2, and the conclusion holds. □

E. Performance Analysis

1) Privacy budget: As shown in Eq. (11), ϵ, which is the constant scale of the privacy budget ϵd, can be determined by γ and $r_{\max}$, as shown in Fig. 2. As we can see, for $r_{\max}$ from 1 to 10^4 and γ from 1 to 10^{-16}, ϵ is always less than 2. Moreover, since ϵ is reversely proportional to $r_{\max}$, the whole privacy budget ϵd is less than $-\log(\gamma)$ (because $d \leq r_{\max}$), which is relatively small. The small privacy budget is critical for differential privacy mechanisms because a large budget result in a large privacy loss.

2) Break the Curse of Dimensionality: As stated in Section III-C, the original ϵ-differential privacy notion and the corresponding Laplacian mechanism suffer from the curse of dimensionality. The reason is that the noise strength added to each element in the text vector has a scale of $S(f_I)/\epsilon$, where $S(f_I)$ is the L_1 sensitivity of the text vector and proportional to the dimension m. We now estimate the scale of the noise strength for the mechanism $K_r(f_I(\cdot))$.

Theorem 4. *Given a text vector $U \in \mathbb{R}^m$, by applying the mechanism $K_r(f_I(\cdot))$, the expected noise strength for each element in U is unrelated to the dimension m.*

Proof. Please check our full version [23]. □

Moreover, by setting the γ to extremely small (e.g., 10^{-16}), the upper bound approaches 1. Note that according to Theorem 1, this upper bound is rather loose. Therefore, the expected noise strength for each element is far less.

3) (α, δ)-usefulness: The mechanism $K_r(f_I(\cdot))$ also satisfies (α, δ)-usefulness defined by [24].

Definition 3 ((α, δ)-usefulness)**.** *A ϵ-text indistinguishability mechanism K satisfies (α, δ)-usefulness iff for every user text vector U, with the probability at least δ, the perturbed text vector U' satisfies $d \leq \alpha$.*

Theorem 5. *The mechanism $K_r(f_I(\cdot))$ defined above achieves the (α, δ)-usefulness.*

Proof. It can be easily seen from the CDF of d in Eq. (8). □

IV. Evaluation

In this section, we use both a real-world dataset and simulations to evaluate the proposed ϵ-text indistinguishability mechanism $K_r(f_I(\cdot))$ in three aspects: the privacy and usefulness, the utility on a typical ML task, and the defense against user-linkage attacks.

A. Dataset

As stated before, the data consumer aims to use the social media data to do the demographics analysis. Here we use a ground truth Twitter user dataset with known age information similar to [5]. Specifically, a user A has age x if one of his friends has posted a tweet with the format "Happy x-birthday to A". We used Twitter Streaming API to monitor

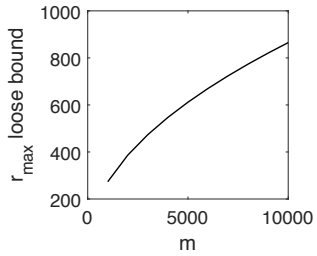


Fig. 3: The loose upper bound of r_{max}.

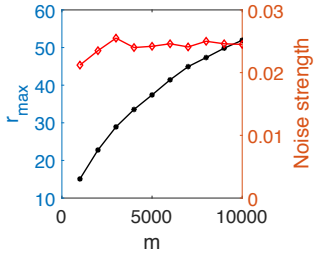
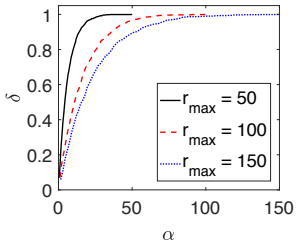
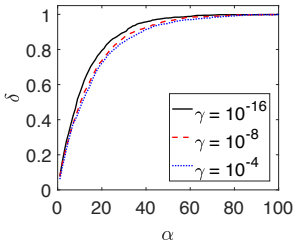


Fig. 4: The real r_{max} and the noise strength for each element.

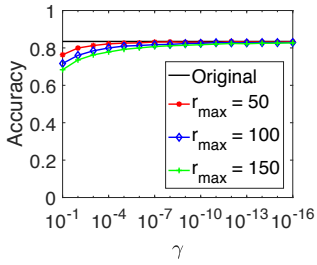


(a) $\gamma = 10^{-8}$

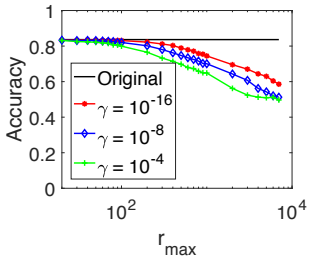


(b) $r_{max} = 100$

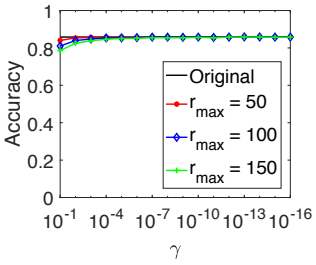
Fig. 5: The usefulness of the mechanism.



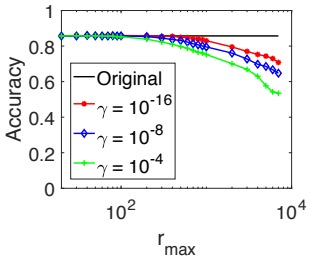
(a) $m = 1000$



(b) $m = 1000$



(c) $m = 5000$



(d) $m = 5000$

Fig. 6: The performance of classification.

these tweets and the ground-truth users. We then manually check the consistency between the claimed age information and the tweets they have posted to. Finally, we found 5,710 users, which consist of 2,855 users who are at least and less than 25 years old, respectively. We crawled one year of their recent tweets and obtained 3,363,706 tweets. We then removed the stopping words and conducted the stemming as stated in Section III-A, and built the TF-IDF matrix according to Eq. (1) for the following experiments. Because of the randomness during the noise generation, we run each of the experiment 100 times and report the average results.

B. Privacy and Usefulness

We first check r_{max} in the real-world dataset above. Fig. 3 shows the loose upper bound with different dimension m stated in Theorem 1. We set the number of users $n = 5710$. The upper bound is a sublinear function with m, and increases from 200 to 1000 when m ranges from 1000 to 10000.

We also measure r_{max} in the dataset as shown in Fig. 4. Specifically, we compute r_{max} as the maximum L_2 norm of each row vector from the dataset $\mathcal{D}$. As we can see, although the r_{max} increases sublinearly with m, it is much less than the upper bound in Fig. 3. The reason is twofold. First, as we built the TF-IDF dataset by choosing the most m frequent grams, the IDF term in Eq.(1) is much less than $\log(n)$. Second, the TF part is less than $\sqrt{m}$ as the text vector is sparse (each user has only used limited grams when m is large).

Given r_{max}, Fig. 4 demonstrates that the expected noise strength added for each single element in the text vector is fairly stable with the dimension m, which is consistent with Theorem 4. Moreover, the expected noise strength ranges from 0.02 to 0.03, and is comparable to the original data. Therefore, the proposed mechanism can tolerate an arbitrary dimension, i.e., breaking the curse of dimensionality.

Fig. 5a and Fig. 5b show the (α, δ)-usefulness of the mechanism at different r_{max} and γ, respectively. As we can see, with probability δ, the distance of the original and perturbed text vector is within α, which verifies Theorem 5.

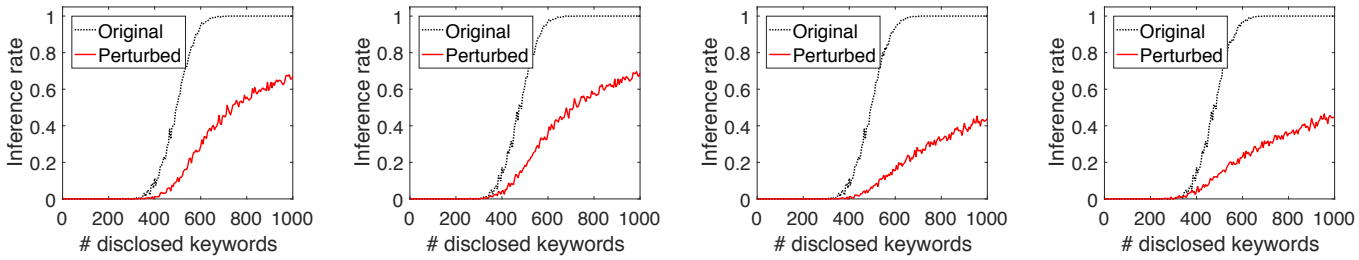
C. Performance on Classification

We evaluate the mechanism on classification, one of the typical applications from the machine learning community. As stated before, each user has the ground-truth age information. We can then build a binary classifier to determine whether a user is younger than 25 years old or not. We use the SVM algorithm to evaluate the performance on both the original and the perturbed datasets by ten-fold cross validation.

Fig. 6a demonstrate the accuracy with γ. The straight and crooked lines represent the original and perturbed datasets, respectively. As we can see, the smaller γ, the higher the performance for the perturbation mechanism. This result is expected as Theorem 4 indicates that the smaller γ, the less the noise added to the original dataset. However, small γ will increase the privacy budget scale ϵ and hence the privacy loss.

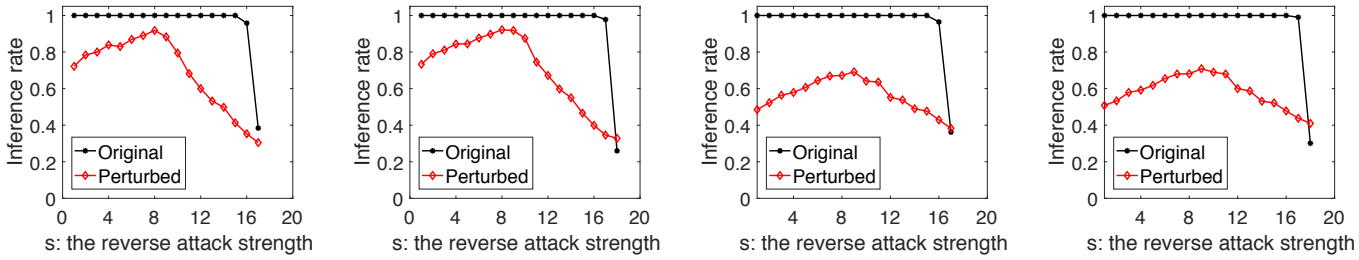
Fig. 6b demonstrates the accuracy of the original dataset (straight line) and the perturbed datasets with r_{max} (crooked curves). It shows that the smaller r_{max}, the better the accuracy because smaller r_{max} will incur less noise. However, less noise will cause a high privacy loss because the attacker can infer the victim given the huge difference of two perturbed vectors.

Fig. 6c and Fig. 6d show the classification performance on $m = 5000$. As we can see, both figures show the similar trend for $m = 1000$, meaning that the mechanism works well at various dimensions. Moreover, the performance when $m =$



(a) $K = 1, r_{max} = 100$ (b) $K = 10, r_{max} = 100$ (c) $K = 1, r_{max} = 200$ (d) $K = 10, r_{max} = 200$

Fig. 7: The performance of inference attack I.



(a) $K = 1, r_{max} = 100$ (b) $K = 10, r_{max} = 100$ (c) $K = 1, r_{max} = 200$ (d) $K = 10, r_{max} = 200$

Fig. 8: The performance of inference attack II.

5000 is slightly better than that when $m = 1000$. The reason is that more keywords lead to better classification.

D. Defense Against User-Linkage Attacks

Our mechanism is designed to defend against the user-linkage attack. The definition of ϵ-text indistinguishability in Definition 2 and the corresponding mechanism in Alg. 1 show that any user can be perturbed to other text vector with certain probability. Therefore, the perturbation can make the user-linkage attack more difficult to conduct. To evaluate the effectiveness of our mechanism, we need to model the strength of the attacker in terms of user inference. We consider two attack models here.

In *inference attack I*, we assume that the attack knows t elements of the victim's text vector, and t vary from 0 to m. We then build an estimated vector U' by keeping these t elements and setting other unknown elements to zero, and check whether the estimated vector U' is in the K-nearest vector set in both the original $\mathcal{D}$ and the perturbed $\mathcal{D}'$. It is expected that the larger the t, the stronger the attack, the higher the inference rate. We set $m = 1000$ and $\gamma = 10^{-8}$. We conduct the experiment by 1000 times and report the average.

Fig. 7 shows the inference rate among the 1-nearest and 10-nearest vectors for $r_{max} = 100$ and 200. We can make two observations. First, all the four curves show that the perturbation makes the user linkage attack much more difficult. Specifically, when the t increase from 300 to 600, the inference rate increase quickly from 0 to 100% for the original dataset. The inference rate then stay at approximate 100% when t is larger than 600. By comparison, the inference rate for the perturbed dataset is at most 68.8% for $r_{max} = 100$ ($K = 10$) and 44.6% for $r_{max} = 200$ ($K = 10$), respectively. Second, Fig. 7 demonstrate the tradeoff between the privacy and usefulness for r_{max}. Specifically, on the one hand, the mechanism's inference rate for $r_{max} = 200$ is less than the rate for $r_{max} = 100$ because larger r_{max} results in larger noise and hence higher-level privacy protection. On the other hand, larger r_{max} results in lower classification performance as indicated in Fig. 6. The tradeoff also holds for γ.

Moreover, users' privacy has not largely sacrifice the utility. For example, as a typical setting, when $r_{max} = 100$ and $\gamma = 10^{-8}$, the inference rate with $t = 600$ and $K = 10$ is 35.9%, and the classification accuracy is reduced by only 1.61%. Therefore, the mechanism can achieve high privacy with little utility loss.

In *inference attack II*, we assume that the attack knows the noisy but the whole text vector of the victim. To that end, we randomly select a victim vector U^* from $\mathcal{D}$, add a noise vector N with the magnitude s where $1/s$ is the attack strength, and then check whether the noisy vector $\tilde{U} = U^* + N$ is in the K-nearest vector set in both the original $\mathcal{D}$ and the perturbed $\mathcal{D}'$. We use the Euclidean distance to represent the difference between any vector pair. Obviously, it is expected that the weaker the attack strength, the higher the inference rate.

Fig. 8 show the inference rate among the 1-nearest and 10-nearest vectors for $r_{max} = 100$ and 200. We can make the similar observations as in the inference attack I. First, the perturbation algorithm makes the user linkage attack much more difficult. Specifically, when the reverse attack strength s increases, the inference rate for the perturbed dataset decreases to about 30% for $K = 1$ and 40% for $K = 10$, meaning that the attacker has limited power to infer the victim. By comparison, the inference rate for the original dataset is always 100% when s is less than 17. The reason is each user text

vector is very distinguishable. When $s > 17$, the inference for the original dataset decreases dramatically because the measured r_{max} for this dataset is 15.1 for $m = 1000$, as indicated in Fig. 4. Second, Fig. 7 demonstrate the tradeoff between the privacy and usefulness in terms of r_{max}, and users' privacy has not largely sacrifice the utility. For example, as a typical setting, when $r_{max} = 100$ and $\gamma = 10^{-8}$, the inference rate with $s = 15$ and $K = 10$ is 47.7%, and the classification accuracy is reduced by only 1.61%.

Note that there is a peak point for the inference rate on the perturbed dataset in Fig. 8. This is because that the perturbation also adds the noise vector in the similar way as in the inference attack II. For different r_{max}, the perturbed vectors have different Euclidean distance from the original vectors. Recall that U' and $\tilde{U}$ are the perturbed vector and the estimated vector from the attacker for the victim U^*, respectively. When the difference of $d(U^*, \tilde{U})$ and $d(U^*, U')$ is small, the inference rate will increase. However, in reality, the attacker has little knowledge on the whole text vector for the victim, and it is difficult to conduct this type of inference.

V. Related Work

Social media platforms host both network and text information, of which the privacy threats both have been widely studied. For the privacy threat from network information, existing results show that an anonymous social graph can be de-anonymized by seed information [25], [26], knowledge graph [27], and the community structures [28]. As for the privacy threat from text information, sophisticated machine learning algorithms can be used to infer a lot of sensitive information, such as age [4], [5], location [6], [7], language [8], and political preference [9].

On the defense side, the research community only attempts to protect user privacy from the perspective of network information. The research efforts fall into two directions. The first line of research [10], [11] aims at protecting vertex privacy by outsourcing social graphs with anonymized user IDs, and the research effort is to prevent the adversary from linking anonymized IDs to corresponding real IDs in the real social network. The other line of research targets link/edge privacy, and the research effort is to outsource social graphs with real user IDs but perturbed edges by outsourcing an obfuscated social network to protect users' privacy [12]–[14]. Our paper is the first to protect the privacy from the text information and is complementary to these efforts.

Privacy-preserving data outsourcing has been thoroughly studied and surveyed in [29]. These techniques such as such as k-anonymity and l-diversity focus on the traditional database and cannot handle unstructured social media data.

Acknowledgement

This work was supported by US Army Research Office (W911NF-15-1-0328), Defense Advanced Research Projects Agency (N66001-17-2-4031) and National Science Foundation (CNS-1619251, CNS-1514381, CNS-1421999, CNS-1320906, CNS-1700032, CNS-1700039, CNS-1651954 (CAREER), CNS-1718078, IIS-1657196, IIS-1718840).

References

[1] P. Gadkari, "How does twitter make money?" Nov. 2013. [Online]. Available: http://www.bbc.com/news/business-24397472

[2] H. Mao, X. Shuai, and A. Kapadia, "Loose tweets: An analysis of privacy leaks on twitter," in *WPES*, Chicago, IL, Octo. 2011.

[3] Twitter, "Twitter api." [Online]. Available: https://dev.twitter.com/rest/public

[4] R. Dey, C. T., K. R., and N. S., "Estimating age privacy leakage in online social networks," in *INFOCOM*, 2012.

[5] J. Zhang, X. Hu, Y. Zhang, and H. Liu, "Your age is no secret: Inferring microbloggers' ages via content and interaction analysis," in *ICWSM*, Cologne, Germany, May 2016.

[6] R. Li, S. Wang, H. Deng, R. Wang, and K. Chang, "Towards social user profiling: Unified and discriminative influence model for inferring home locations," in *KDD*, Beijing, China, Aug. 2012.

[7] J. Zhang, J. Sun, R. Zhang, and Y. Zhang, "Your actions tell where you are: Uncovering twitter users in a metropolitan area," in *IEEE CNS*, Florence, Italy, Sep. 2015.

[8] D. Nguyen, R. Gravel, D. Trieschnigg, and T. Meder, ""how old do you think i am?"; a study of language and age in twitter," in *ICWSM*, Boston, IL, Jul. 2013.

[9] X. Chen, Y. Wang, E. Agichtein, and F. Wang, "A comparative study of demographic attribute inference in twitter," in *ICWSM*, Oxford, England, May 2015.

[10] C.-H. Tai, P.-J. Tseng, P. S. Yu, and M.-S. Chen, "Identities anonymization in dynamic social networks," in *ICDM*.

[11] G. Wang, Q. Liu, F. Li, S. Yang, and J. Wu, "Outsourcing privacy-preserving social networks to a cloud," in *INFOCOM*, 2013.

[12] P. Mittal, C. Papamanthou, and D. Song, "Preserving link privacy in social network based systems," in *NDSS*, San Diago, CA, Feb. 2013.

[13] C. Liu and P. Mittal, "Linkmirage: How to anonymize links in dynamic social systems," in *NDSS*, San Diago, CA, Feb. 2016.

[14] F. Ahmed, A. X. Liu, and R. Jin, "Social graph publishing with privacy guarantees," in *ICDCS*, 2016.

[15] C. Dwork, "Differential privacy," in *Automata, languages and programming*.

[16] R. Bellman, *Dynamic Programming*. Dover Publications, 2003.

[17] "An exhaustive study of twitter users across the world," Oct. 2012. [Online]. Available: http://temp.beevolve.com/twitter-statistics/

[18] R. Dey, Z. Jelveh, and K. Ross, "Facebook users have become much more private: A large-scale study," in *IEEE PERCOM Workshops*, 2012.

[19] M. Andrés, N. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *CCS*, Berlin, Germany, 2013.

[20] M. Porter, *Readings in information retrieval*. Morgan Kaufmann Publishers Inc., 1997, ch. An algorithm for suffix stripping, pp. 313–316.

[21] J. Leskovec, A. Rajaraman, and J. Ullman, *Mining Massive Datasets*. Cambridge University Press, 2014, ch. Data Mining, pp. 7–9.

[22] C. Liu, S. Chakraborty, and P. Mittal, "Dependence makes you vulnerable: Differential privacy under dependent tuples," in *NDSS*, San Diego, CA, Feb. 2016.

[23] J. Zhang, J. Sun, R. Zhang, Y. Zhang, and X. Hu, "Privacy-preserving social media data outsourcing." [Online]. Available: http://cnsg.asu.edu/papers/jxzhangINFOCOM18Full.pdf

[24] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to non-interactive database privacy," in *STOC*, Victoria, British Columbia, Canada, 2008.

[25] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *SP*, 2009.

[26] S. Ji, W. Li, N. Z. Gong, P. Mittal, and R. Beyah, "On your social network de-anonymizablity: Quantification and large scale evaluation with seed knowledge." in *NDSS*, San Diago, CA, Feb. 2015.

[27] J. Qian, X. Y. Li, C. Zhang, and L. Chen, "De-anonymizing social networks and inferring private attributes using knowledge graphs," in *INFOCOM*, 2016.

[28] S. Nilizadeh, A. Kapadia, and Y.-Y. Ahn, "Community-enhanced de-anonymization of online social networks," in *CCS*, Scottsdale, AZ, 2014.

[29] B. Fung, K. Wang, R. Chen, and P. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys (CSUR)*, vol. 42, no. 4, pp. 14:1–14:52, 2010.