# Towards Privacy for Social Networks: A Zero-Knowledge Based Definition of Privacy

Johannes Gehrke, Edward Lui, and Rafael Pass*

Cornell University
{johannes,luied,rafael}@cs.cornell.edu

**Abstract.** We put forward a zero-knowledge based definition of privacy. Our notion is strictly stronger than the notion of differential privacy and is particularly attractive when modeling privacy in social networks. We furthermore demonstrate that it can be meaningfully achieved for tasks such as computing averages, fractions, histograms, and a variety of graph parameters and properties, such as average degree and distance to connectivity. Our results are obtained by establishing a connection between zero-knowledge privacy and sample complexity, and by leveraging recent sublinear time algorithms.

## 1 Introduction

Data privacy is a fundamental problem in today's information age. Enormous amounts of data are collected by government agencies, search engines, social networking systems, hospitals, financial institutions, and other organizations, and are stored in databases. There are huge social benefits in analyzing this data; however, it is important that sensitive information about individuals who have contributed to the data is not leaked to users analyzing the data. Thus, one of the main goals is to release statistical information about the population who have contributed to the data without breaching their individual privacy.

Many privacy definitions and schemes have been proposed in the past (see [4] and [11] for surveys). However, many of them have been shown to be insufficient by describing realistic attacks on such schemes (e.g., see [19]). The notion of *differential privacy* [8, 7], however, has remained strong and resilient to these attacks. Differential privacy requires that when one person's data is added or removed from the database, the output of the database access mechanism changes very little so that the output before and after the change are "$\epsilon$-close" (where a specific notion of closeness of distributions is used). This notion has quickly become the standard notion of privacy, and mechanisms for releasing a variety of functions (including histogram queries, principal component analysis, learning, and many more (see [6] for a recent survey)) have been developed.

* Pass is supported in part by a Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR Award F56-8414, BSF Grant 2006317 and I3P grant 2006CS-001-0000001-02.

As we shall argue, however, although differential privacy provides a strong privacy guarantee, there are realistic social network settings where these guarantees might not be strong enough. Roughly speaking, the notion of differential privacy can be rephrased as requiring that whatever an adversary learns about an individual could have been recovered about the individual had the adversary known every other individual in the database (see the appendix of [8] for a formalization of this statement). Such a privacy guarantee is not sufficiently strong in the setting of social networks where an individual's *friends* are strongly correlated with the individual; in essence, "if I know your friends, I know you". (Indeed, a recent study [17] indicates that an individual's sexual orientation can be accurately predicted just by looking at the person's Facebook friends.) We now give a concrete example to illustrate how a differentially private mechanism can violate the privacy of individuals in a social network setting.

*Example 1 (Democrats vs. Republicans).* Consider a social network of $n$ people that are grouped into cliques of size 200. In each clique, either at least 80% of the people are Democrats, or at least 80% are Republicans. However, assume that the number of Democrats overall is roughly the same as the number of Republicans. Now, consider a mechanism that computes the proportion (in $[0, 1]$) of Democrats in each clique and adds just enough Laplacian noise to satisfy $\epsilon$-differential privacy for a small $\epsilon$, say $\epsilon = 0.1$. For example, to achieve $\epsilon$-differential privacy, it suffices to add $Lap(\frac{1}{200\epsilon})$ noise[1] to each clique independently, since if a single person changes his or her political preference, the proportion for the person's clique changes by $\frac{1}{200}$ (see Proposition 1 in [8]).

Since the mechanism satisfies $\epsilon$-differential privacy for a small $\epsilon$, one may think that it is safe to release such information without violating the privacy of any particular person. That is, the released data should not allow us to guess correctly with probability significantly greater than $\frac{1}{2}$ whether a particular person is a Democrat or a Republican. However, this is not the case. With $\epsilon = 0.1$, $Lap(\frac{1}{200\epsilon})$ is a small amount of noise, so with high probability, the data released will tell us the main political preference for any particular clique. An adversary that knows which clique a person is in will be able to correctly guess the political preference of that person with probability close to 80%.

*Remark 1.* In the above example, we assume that the graph structure is known and that the adversary can identify what clique an individual is in. Such information is commonly available: Graph structures of (anonymized) social networks are often released; these may include a predefined or natural clustering of the people (nodes) into cliques. Furthermore, an adversary may often also figure out the identity of various nodes in the graph (see [1, 16]); in fact, by participating in the social network before the anonymized graph is published, an adversary can even target specific individuals of his or her choice (see [1]).

Differential privacy says that the output of the mechanism does not depend much on any particular individual's data in the database. Thus, in the above example, a person has little reason not to truthfully report his political preference.

[1] $Lap(\lambda)$ is the Laplace distribution with probability density function $f_\lambda(x) = \frac{1}{2\lambda} e^{\frac{|x|}{\lambda}}$.

However, this does not necessarily imply that the mechanism does not violate the person's privacy. In situations where a social network provides auxiliary information about an individual, that person's privacy can be violated even if he decides to not have his information included!

It is already known that differential privacy may not provide a strong enough privacy guarantee when an adversary has specific auxiliary information about an individual. For example, it was pointed out in [7] that if an adversary knows the auxiliary information "person A is two inches shorter than the average American woman", and if a differentially private mechanism accurately releases the average height of American women, then the adversary learns person A's height (which is assumed to be sensitive information in this example). In this example, the adversary has very specific auxiliary information about an individual that is usually hard to obtain. However, in the Democrats vs. Republicans example, the auxiliary information (the graph and clique structure) about individuals is more general and more easily accessible. Since social network settings contain large amounts of auxiliary information and correlation between individuals, differential privacy is usually not strong enough in such settings.

One may argue that there are versions of differential privacy that protect the privacy of groups of individuals, and that the mechanism in the Democrats vs. Republicans example does not satisfy these stronger definitions of privacy. While this is true, the main point here is that differential privacy will not protect the privacy of an individual, even though the definition is designed for individual privacy. Furthermore, even if we had used a differentially private mechanism that ensures privacy for groups of size 200 (i.e., the size of each clique), it might still be possible to deduce information about an individual by looking at the *friends of the friends* of the individual; this includes a significantly larger number of individuals.[2]

### 1.1 Towards a Zero-Knowledge Definition of Privacy

In 1977, Dalenius [5] stated a privacy goal for statistical databases: anything about an individual that can be learned from the database can also be learned without access to the database. This would be a very desirable notion of privacy. Unfortunately, Dwork and Naor [7, 9] demonstrated a general impossibility result showing that a formalization of Dalenius's goal along the lines of semantic security for cryptosystems cannot be achieved, assuming that the database gives any non-trivial utility.

Our aim is to provide a privacy definition along the lines of Dalenius, and more precisely, relying on the notion of zero-knowledge from cryptography. In this context, the traditional notion of zero-knowledge says that an adversary gains essentially "zero additional knowledge" by accessing the mechanism. More precisely, whatever an adversary can compute by accessing the mechanism can essentially also be computed without accessing the mechanism. A mechanism

[2] The number of "friends of friends" is usually larger than the square of the number of friends (see [23]).

satisfying this property would be private but utterly useless, since the mechanism provides essentially no information. The whole point of releasing data is to provide utility; thus, this extreme notion of zero-knowledge, which we now call "complete zero-knowledge", is not very applicable in this setting.

Intuitively, we want the mechanism to not release any additional information beyond some *"aggregate information"* that is considered acceptable to release. To capture this requirement, we use the notion of a "simulator" from zero-knowledge, and we require that a simulator with the acceptable aggregate information can essentially compute whatever an adversary can compute by accessing the mechanism. Our zero-knowledge privacy definition is thus stated relative to some class of algorithms providing acceptable aggregate information.

**Aggregate Information** The question is how to define appropriate classes of aggregate information. We focus on the case where the aggregate information is any information that can be obtained from $k$ random samples/rows (each of which corresponds to one individual's data) of the database, where the data of the person the adversary wants to attack has been concealed. The value of $k$ can be carefully chosen so that the aggregate information obtained does not allow one to infer (much) information about the concealed data. The simulator is given this aggregate information and has to compute what the adversary essentially computes, even though the adversary has access to the mechanism. This ensures that the mechanism does not release any additional information beyond this "$k$ random sample" aggregate information given to the simulator.

Differential privacy can be described using our zero-knowledge privacy definition by considering simulators that are given aggregate information consisting of the data of all but one individual in the database; this is the same as aggregate information consisting of "$k$ random samples" with $k = n$, where $n$ is the number of rows in the database (recall that the data of the individual the adversary wants to attack is concealed), which we formally prove later. For $k$ less than $n$, such as $k = \log n$ or $k = \sqrt{n}$, we obtain notions of privacy that are stronger than differential privacy. For example, we later show that the mechanism in the Democrats vs. Republicans example does not satisfy our zero-knowledge privacy definition when $k = o(n)$ and $n$ is sufficiently large.

We may also consider more general models of aggregate information that are specific to graphs representing social networks; in this context we focus on random samples with some exploration of the neighborhood of each sample.

### 1.2 Our Results

We consider two different settings for releasing information. In the first setting, we consider statistical (row) databases in a setting where an adversary might have auxiliary information, such as from a social network, and we focus on releasing traditional statistics (e.g., averages, fractions, histograms, etc.) from a database. As explained earlier, differential privacy may not be strong enough in such a setting, so we use our zero-knowledge privacy definition instead. In the

second setting, we consider graphs with personal data that represent social networks, and we focus on releasing information directly related to a social network, such as properties of the graph structure.

*Setting #1. Computing functions on databases with zero-knowledge privacy:* In this setting, we focus on computing functions mapping databases to $\mathbb{R}^m$. Our main result is a characterization of the functions that can be released with zero-knowledge privacy in terms of their *sample complexity*—i.e., how accurate the function can be approximated using random samples from the input database. More precisely, functions with low sample complexity can be computed accurately by a zero-knowledge private mechanism, and vice versa. It is already known that functions with low sample complexity can be computed with differential privacy (see [8]), but here we show that the stronger notion of zero-knowledge privacy can be achieved. In this result, the zero-knowledge private mechanism we construct simply adds Laplacian noise appropriately calibrated to the sample complexity of the function.

Many common queries on statistical databases have low sample complexity, including averages, sum queries, and coarse histogram queries. (In general, it would seem that any "meaningful" query function for statistical databases should have relatively low sample complexity if we think of the rows of the database as random samples from some large underlying population). As a corollary of our characterization we get zero-knowledge private mechanisms for all these functions providing decent utility guarantees. These results can be found in Section 3.

*Setting #2. Releasing graph structure information with zero-knowledge privacy:* In this setting, we consider a graph representing a social network, and we focus on privately releasing information about the structure of the graph. We use our zero-knowledge privacy definition, since the released information can be combined with auxiliary information such as an adversary's knowledge and/or previously released data (e.g., graph structure information) to breach the privacy of individuals.

The connection between sample complexity and zero-knowledge privacy highlights an interesting connection between *sublinear time algorithms* and privacy. As it turns out, many of the recently developed sublinear algorithms on graphs proceed by picking random samples (and next performing some local exploration); we are able to leverage these algorithms to privately release graph structure information, such as average degree and distance to properties such as connectivity and cycle-freeness. We discuss these results in Section 4.

## 2 Zero-Knowledge Privacy

### 2.1 Definitions

Let $\mathcal{D}$ be the class of all databases whose rows are tuples from some relation/universe $X$. For convenience, we will assume that $X$ contains a tuple $\perp$,

which can be used to conceal the true value of a row. Given a database $D$, let $|D|$ denote the number of rows in $D$. For any integer $n$, let $[n]$ denote the set $\{1, \ldots, n\}$. For any database $D \in \mathcal{D}$, any integer $i \in [|D|]$, and any $v \in X$, let $(D_{-i}, v)$ denote the database $D$ with row $i$ replaced by the tuple $v$.

In this paper, mechanisms, adversaries, and simulators are simply randomized algorithms that play certain roles in our definitions. Let $San$ be a mechanism that operates on databases in $\mathcal{D}$. For any database $D \in \mathcal{D}$, any adversary $A$, and any $z \in \{0,1\}^*$, let $Out_A(A(z) \leftrightarrow San(D))$ denote the random variable representing the output of $A$ on input $z$ after interacting with the mechanism $San$ operating on the database $D$. Note that $San$ can be interactive or non-interactive. If $San$ is non-interactive, then $San(D)$ sends information (e.g., a sanitized database) to $A$ and then halts immediately; the adversary $A$ then tries to breach the privacy of some individual in the database $D$.

Let $agg$ be any class of randomized algorithms that provide aggregate information to simulators, as described in Section 1.1. We refer to $agg$ as a *model of aggregate information*.

**Definition 1.** *We say that San is* $\epsilon$**-zero-knowledge private with respect to** $agg$ *if there exists a* $T \in agg$ *such that for every adversary* $A$*, there exists a simulator* $S$ *such that for every database* $D \in X^n$*, every* $z \in \{0,1\}^*$*, every integer* $i \in [n]$*, and every* $W \subseteq \{0,1\}^*$*, the following hold:*

- $\Pr[Out_A(A(z) \leftrightarrow San(D)) \in W] \leq e^\epsilon \cdot \Pr[S(z, T(D_{-i}, \perp), i, n) \in W]$
- $\Pr[S(z, T(D_{-i}, \perp), i, n) \in W] \leq e^\epsilon \cdot \Pr[Out_A(A(z) \leftrightarrow San(D)) \in W]$

*The probabilities are over the random coins of San and A, and T and S, respectively.*

Intuitively, the above definition says that whatever an adversary can compute by accessing the mechanism can essentially also be computed without accessing the mechanism but with certain aggregate information (specified by agg). The adversary in the latter scenario is represented by the simulator $S$. The definition requires that the adversary's output distribution is close to that of the simulator. This ensures that the mechanism essentially does not release any additional information beyond what is allowed by $agg$. When the algorithm $T$ provides aggregate information to the simulator $S$, the data of individual $i$ is concealed so that the aggregate information does not depend directly on individual $i$'s data. However, in the setting of social networks, the aggregate information may still depend on people's data that are correlated with individual $i$ in reality, such as the data of individual $i$'s friends. Thus, the role played by $agg$ is very important in the context of social networks.

To measure the closeness of the adversary's output and the simulator's output, we use the same closeness measure as in differential privacy (as opposed to, say, statistical difference) for the same reasons. As explained in [8], consider a mechanism that outputs the contents of a randomly chosen row. Suppose $agg$ is defined so that it includes the algorithm that simply outputs its input $(D_{-i}, \perp)$ to the simulator (which is the case of differential privacy; see Section 1.1 and

2.2). Then, a simulator can also choose a random row and then simulate the adversary with the chosen row sent to the simulated adversary. The real adversary's output will be very close to the simulator's output in statistical difference ($1/n$ to be precise); however, it is clear that the mechanism always leaks private information about some individual.

*Remark 2.* Our $\epsilon$-zero-knowledge privacy definition can be easily extended to $(\epsilon, \delta(\cdot))$-zero-knowledge privacy, where we also allow an additive error of $\delta(n)$ on the RHS of the inequalities. We can further extend our definition to $(c, \epsilon, \delta(\cdot))$-zero-knowledge privacy to protect the privacy of any group of $c$ individuals simultaneously. To obtain this more general definition, we would change "$i \in [n]$" to "$I \subseteq [n]$ with $|I| \leq c$", and "$S(z, (D_{-i}, \perp), i, n)$" to "$S(z, (D_{-I}, \perp), I, n)$". We use this more general definition when we consider group privacy.

*Remark 3.* In our zero-knowledge privacy definition, we consider computationally unbounded simulators. We can also consider PPT simulators by requiring that the mechanism $San$ and the adversary $A$ are PPT algorithms, and $agg$ is a class of PPT algorithms. All of these algorithms would be PPT in $n$, the size of the database. With minor modifications, the results of this paper would still hold in this case.

The choice of $agg$ determines the type and amount of aggregate information given to the simulator, and should be decided based on the context in which the zero-knowledge privacy definition is used. The aggregate information should not depend much on data that is highly correlated with the data of a single person, since such aggregate information may be used to breach the privacy of that person. For example, in the context of social networks, such aggregate information should not depend much on any person and the people closely connected to that person, such as his or her friends. By choosing $agg$ carefully, we ensure that the mechanism essentially does not release any additional information beyond what is considered acceptable. We first consider the model of aggregate information where $T$ in the definition of zero-knowledge privacy chooses $k(n)$ random samples. Let $k : \mathbb{N} \to \mathbb{N}$ be any function.

- $RS(k(\cdot)) = k(\cdot)$ random samples: the class of algorithms $T$ such that on input a database $D \in X^n$, $T$ chooses $k(n)$ random samples (rows) from $D$ uniformly without replacement, and then performs any computation on these samples without reading any of the other rows of $D$. Note that with such samples, $T$ can emulate choosing $k(n)$ random samples with replacement, or a combination of without replacement and with replacement.

$k(n)$ should be carefully chosen so that the aggregate information obtained does not allow one to infer (much) information about the concealed data. For $k(n) = 0$, the simulator is given no aggregate information at all, which is the case of complete zero-knowledge. For $k(n) = n$, the simulator is given all the rows of the original database except for the target individual $i$, which is the case of differential privacy (as we prove later). For $k(n)$ strictly in between 0 and

$n$, we obtain notions of privacy that are stronger than differential privacy. For example, one can consider $k(n) = o(n)$, such as $k(n) = \log n$ or $k(n) = \sqrt{n}$.

In the setting of a social network, $k(n)$ can be chosen so that when $k(n)$ random samples are chosen from $(D_{-i}, \perp)$, with very high probability, for (almost) all individuals $j$, very few of the $k(n)$ chosen samples will be in individual $j$'s local neighborhood in the social network graph. This way, the aggregate information released by the mechanism depends very little on data that is highly correlated with the data of a single individual. The choice of $k(n)$ would depend on various properties of the graph structure, such as clustering coefficient, edge density, and degree distribution. The choice of $k(n)$ would also depend on the amount of correlation between the data of adjacent or close vertices (individuals) in the graph, and the type of information released by the mechanism. In this model of aggregate information, vertices (individuals) in the graph with more adjacent vertices (e.g., representing friends) may have less privacy than those with fewer adjacent vertices. However, this is often the case in social networks, where having more links/connections to other people may result in less privacy.

In the remainder of this section, we focus primarily on the $RS(k(\cdot))$ model of aggregate information. In Section 4, we consider other models of aggregate information that take more into consideration the graph structure of a social network. Note that zero-knowledge privacy does not necessarily guarantee that the privacy of every individual is completely protected. Zero-knowledge privacy is defined with respect to a model of aggregate information, and such aggregate information may still leak some sensitive information about an individual in certain scenarios.

*Composition:* Just as for differentially private mechanisms, mechanisms that are $\epsilon$-zero-knowledge private with respect to $RS(k(\cdot))$ also compose nicely.

**Proposition 1.** *Suppose $San_1$ is $\epsilon_1$-zero-knowledge private with respect to $RS(k_1(\cdot))$ and $San_2$ is $\epsilon_2$-zero-knowledge private with respect to $RS(k_2(\cdot))$. Then, the mechanism obtained by composing $San_1$ with $San_2$ is $(\epsilon_1+\epsilon_2)$-zero-knowledge private with respect to $RS((k_1 + k_2)(\cdot))$.*

See the full version of this paper ([12]) for the proof.

*Graceful Degradation for Group Privacy:* A nice feature of differential privacy is that $\epsilon$-differential privacy implies $(c, c\epsilon)$-differential privacy for groups of size $c$ (see [7] and the appendix in [8]). However, the $c\epsilon$ appears in the exponent of $e$ in the definition of $(c, c\epsilon)$-differential privacy, so the degradation is exponential in $c$. Thus, the group privacy guarantee implied by $\epsilon$-differential privacy is not very meaningful unless the group size $c$ is small. We do not have a group privacy guarantee for pure $\epsilon$-zero-knowledge privacy; however, we do have a group privacy guarantee for $(\epsilon, \delta(\cdot))$-zero-knowledge privacy with respect to $RS(k(\cdot))$ that does not degrade at all for $\epsilon$, and only degrades linearly for $\delta(\cdot)$ with increasing group size.

**Proposition 2.** *Suppose San is $(\epsilon, \delta(\cdot))$-zero-knowledge private with respect to $RS(k(\cdot))$. Then, for every $c \geq 1$, San is also $(c, \epsilon, \delta_c(\cdot))$-zero-knowledge private with respect to $RS(k(\cdot))$, where $\delta_c(n) = \delta(n) + e^{\epsilon}(c-1) \cdot \frac{k(n)}{n}$.*

See the full version of this paper for the proof. Intuitively, for $k(n)$ sufficiently smaller than $n$, $(\epsilon, \delta(\cdot))$-zero-knowledge privacy with respect to $RS(k(\cdot))$ actually implies some notion of group privacy, since the algorithm $T$ (in the privacy definition) chooses each row with probability $k(n)/n$. Thus, $T$ chooses any row of a fixed group of $c$ rows with probability at most $ck(n)/n$. If this probability is very small, then the output of $T$ and thus the simulator $S$ does not depend much on any group of $c$ rows.

### 2.2 Differential Privacy vs. Zero-Knowledge Privacy

In this section, we compare differential privacy to our zero-knowledge privacy definition. We first state the definition of differential privacy in a form similar to our zero-knowledge privacy definition in order to more easily compare the two. For any pair of databases $D_1, D_2 \in X^n$, let $H(D_1, D_2)$ denote the number of rows in which $D_1$ and $D_2$ differ, comparing row-wise.

**Definition 2.** *We say that San is* **$\epsilon$-differentially private** *if for every adversary A, every $z \in \{0,1\}^*$, every pair of databases $D_1, D_2 \in X^n$ with $H(D_1, D_2) \leq 1$, and every $W \subseteq \{0,1\}^*$, we have*

$$\Pr[Out_A(A(z) \leftrightarrow San(D_1)) \in W] \leq e^{\epsilon} \cdot \Pr[Out_A(A(z) \leftrightarrow San(D_2)) \in W],$$

*where the probabilities are over the random coins of San and A. For $(c, \epsilon)$-* **differential privacy** *(for groups of size c), the "$H(D_1, D_2) \leq 1$" is changed to "$H(D_1, D_2) \leq c$".*

**Proposition 3.** *Suppose San is $\epsilon$-zero-knowledge private with respect to any class agg. Then, San is $2\epsilon$-differentially private.*

**Proposition 4.** *Suppose San is $\epsilon$-differentially private. Then, San is $\epsilon$-zero-knowledge private with respect to $RS(n)$.*

See the full version of this paper for the proof of Propositions 3 and 4.

*Remark 4.* If we consider PPT simulators in the definition of zero-knowledge privacy instead of computationally unbounded simulators, then we require *San* in Proposition 4 to be PPT as well.

Combining Propositions 3 and 4, we see that our zero-knowledge privacy definition includes differential privacy as a special case (up to a factor of 2 for $\epsilon$).

### 2.3 Revisiting the Democrats vs. Republicans Example

Recall the Democrats vs. Republicans example in the introduction. The mechanism in the example is $\epsilon$-differentially private for some small $\epsilon$, even though the privacy of individuals is clearly violated. However, the mechanism is not zero-knowledge private in general. Suppose that the people's political preferences are stored in a database $D \in X^n$.

**Proposition 5.** *Fix $\epsilon > 0$, $c \geq 1$, and any function $k(\cdot)$ such that $k(n) = o(n)$. Let San be a mechanism that on input $D \in X^n$ computes the proportion of Democrats in each clique and adds $Lap(\frac{c}{200\epsilon})$ noise to each proportion independently. Then, San is $(c, \epsilon)$-differentially private, but for every sufficiently large $n$, San is not $\epsilon'$-zero-knowledge private with respect to $RS(k(\cdot))$ for any constant $\epsilon' > 0$.*

See the full version of this paper for the proof. Intuitively, the last part of the proposition holds because for sufficiently large $n$, with high probability there exists some clique such that an adversary having only $k(n) = o(n)$ random samples would not have any samples in that clique. Thus, with high probability, there exists some clique that the adversary knows nothing about. Therefore, the adversary does gain knowledge by accessing the mechanism, which gives some information about every clique since the amount of noise added to each clique is constant.

*Remark 5.* In the Democrats vs. Republicans example, even if *San* adds $Lap(\frac{1}{\epsilon})$ noise to achieve $(200, \epsilon)$-differential privacy so that the privacy of each clique (and thus each person) is protected, the mechanism would still fail to be $\epsilon'$-zero-knowledge private with respect to $RS(k(\cdot))$ for any constant $\epsilon' > 0$ when $n$ is sufficiently large (see Proposition 5). Thus, zero-knowledge privacy with respect to $RS(k(\cdot))$ with $k(n) = o(n)$ seems to provide an unnecessarily strong privacy guarantee in this particular example. However, this is mainly because the clique size is fixed and known to be 200, and we have assumed that the only correlation between people's political preferences that exists is within a clique. In a more realistic social network, there would be cliques of various sizes, and the correlation between people's data would be more complicated. For example, an adversary knowing your friends' friends may still be able to infer a lot of information about you.

## 3 Characterizing Zero-Knowledge Privacy

In this section, we focus on constructing zero-knowledge private mechanisms that compute a function mapping databases in $X^n$ to $\mathbb{R}^m$, and we characterize the set of functions that can be computed with zero-knowledge privacy. These are precisely the functions with low sample complexity, i.e., can be approximated (accurately) using only limited information from the database, such as $k$ random samples.

We quantify the error in approximating a function $g : X^n \to \mathbb{R}^m$ using $L_1$ distance. Let the $L_1$-sensitivity of $g$ be defined by $\Delta(g) = \max\{||g(D') - g(D'')||_1 : D', D'' \in X^n \text{ s.t. } H(D', D'') \leq 1\}$. Let $\mathcal{C}$ be any class of randomized algorithms.

**Definition 3.** *A function $g : X^n \to \mathbb{R}^m$ is said to have $(\delta, \beta)$-**sample complexity** with respect to $\mathcal{C}$ if there exists an algorithm $T \in \mathcal{C}$ such that for every input $D \in X^n$, we have $T(D) \in \mathbb{R}^m$ and*

$$\Pr[||T(D) - g(D)||_1 \leq \delta] \geq 1 - \beta.$$

*$T$ is said to be a $(\delta, \beta)$-**sampler** for $g$ with respect to $\mathcal{C}$.*

*Remark 6.* If we consider PPT simulators in the definition of zero-knowledge privacy instead of computationally unbounded simulators, then we would require here that $\mathcal{C}$ is a class of PPT algorithms (PPT in $n$, the size of the database). Thus, in the definition of $(\delta, \beta)$-sample complexity, we would consider a family of functions (one for each value of $n$) that can be computed in PPT, and the sampler $T$ would be PPT in $n$.

It was shown in [8] that functions with low sample complexity with respect to $RS(k(\cdot))$ have low sensitivity as well.

**Lemma 1 ([8]).** *Suppose $g : X^n \to \mathbb{R}^m$ has $(\delta, \beta)$-sample complexity with respect to $RS(k(\cdot))$ for some $\beta < \frac{1-k(n)/n}{2}$. Then, $\Delta(g) \leq 2\delta$.*

As mentioned in [8], the converse of the above lemma is not true, i.e., not all functions with low sensitivity have low sample complexity (see [8] for an example). This should be no surprise, since functions with low sensitivity have accurate differentially private mechanisms, while functions with low sample complexity have accurate zero-knowledge private mechanisms. We already know that zero-knowledge privacy is stronger than differential privacy, as illustrated by the Democrats vs. Republicans example.

We now state how the sample complexity of a function is related to the amount of noise a mechanism needs to add to the function value in order to achieve a certain level of zero-knowledge privacy.

**Proposition 6.** *Suppose $g : X^n \to [a, b]^m$ has $(\delta, \beta)$-sample complexity with respect to some $\mathcal{C}$. Then, the mechanism $San(D) = g(D) + (X_1, \ldots, X_m)$, where $X_j \sim Lap(\lambda)$ for $j = 1, \ldots, m$ independently, is $\ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}})$-zero-knowledge private with respect to $\mathcal{C}$.*

The intuition is that the sampling error gets blurred by the noise added.

*Proof.* Let $T$ be a $(\delta, \beta)$-sampler for $g$ with respect to $\mathcal{C}$. Let $A$ be any adversary. Let $S$ be a simulator that, on input $(z, T(D_{-i}, \perp), i, n)$, first checks whether $T(D_{-i}, \perp)$ is in $[a, b]^m$; if not, $S$ projects $T(D_{-i}, \perp)$ onto the set $[a, b]^m$ (with respect to $L_1$ distance) so that the accuracy of $T(D_{-i}, \perp)$ is improved and

$||g(D)-T(D_{-i},\perp)||_1 \leq (b-a)m$ always holds, which we use later. From here on, $T(D_{-i},\perp)$ is treated as a random variable that reflects the possible modification $S$ may perform. The simulator $S$ computes $T(D_{-i},\perp)+(X_1,\ldots,X_m)$, which we will denote using the random variable $S'(z,T(D_{-i},\perp),i,n)$. $S$ then simulates the computation of $A(z)$ with $S'(z,T(D_{-i},\perp),i,n)$ sent to $A$ as a message, and outputs whatever $A$ outputs.

Let $D \in X^n$, $z \in \{0,1\}^*$, $i \in [n]$. Fix $x \in T(D_{-i},\perp)$ and $s \in \mathbb{R}^m$. Then, we have

$$\begin{aligned}
&\max\left\{\frac{f_\lambda(s-g(D))}{f_\lambda(s-x)}, \frac{f_\lambda(s-x)}{f_\lambda(s-g(D))}\right\} \\
=\ &\max\left\{e^{(\frac{1}{\lambda}\cdot(||s-x||_1-||s-g(D)||_1))}, e^{(\frac{1}{\lambda}\cdot(||s-g(D)||_1-||s-x||_1))}\right\} \\
\leq\ & e^{(\frac{1}{\lambda}\cdot||g(D)-x||_1)} \leq e^{(\frac{1}{\lambda}\cdot(||g(D)-g(D_{-i},\perp)||_1+||g(D_{-i},\perp)-x||_1))} \\
\leq\ & e^{(\frac{1}{\lambda}\cdot(\Delta(g)+||g(D_{-i},\perp)-x||_1))}. \qquad (1)
\end{aligned}$$

Since $||g(D)-x||_1 \leq (b-a)m$ always holds, we also have

$$\max\left\{\frac{f_\lambda(s-g(D))}{f_\lambda(s-x)}, \frac{f_\lambda(s-x)}{f_\lambda(s-g(D))}\right\} \leq e^{(\frac{1}{\lambda}\cdot||g(D)-x||_1)} \leq e^{\frac{(b-a)m}{\lambda}}. \qquad (2)$$

Since $T$ is a $(\delta,\beta)$-sampler for $g$, we have $\Pr[||g(D_{-i},\perp)-T(D_{-i},\perp)||_1 \leq \delta] \geq 1-\beta$. Thus, using (1) and (2) above, we have

$$\ln\left(\frac{\sum_{x\in T(D_{-i},\perp)} f_\lambda(s-x)\cdot\Pr[T(D_{-i},\perp)=x]}{f_\lambda(s-g(D))}\right) \leq \ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}}+\beta e^{\frac{(b-a)m}{\lambda}}).$$

Now, using (1) and (2) again, we also have

$$\begin{aligned}
&\ln\left(\frac{f_\lambda(s-g(D))}{\sum_{x\in T(D_{-i},\perp)} f_\lambda(s-x)\cdot\Pr[T(D_{-i},\perp)=x]}\right) \\
=\ &-\ln\left(\frac{\sum_{x\in T(D_{-i},\perp)} f_\lambda(s-x)\cdot\Pr[T(D_{-i},\perp)=x]}{f_\lambda(s-g(D))}\right) \\
\leq\ &-\ln((1-\beta)e^{-\frac{\Delta(g)+\delta}{\lambda}}+\beta e^{-\frac{(b-a)m}{\lambda}}) = \ln(((1-\beta)e^{-\frac{\Delta(g)+\delta}{\lambda}}+\beta e^{-\frac{(b-a)m}{\lambda}})^{-1}) \\
\leq\ &\ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}}+\beta e^{\frac{(b-a)m}{\lambda}}),
\end{aligned}$$

where the last inequality follows from the fact that the function $f(x)=x^{-1}$ is convex for $x>0$. Then, for every $s \in \mathbb{R}^n$, we have

$$\begin{aligned}
&\left|\ln\left(\frac{\Pr[San(D)=s]}{\Pr[S'(z,T(D_{-i},\perp),i,n)=s]}\right)\right| \\
=\ &\left|\ln\left(\frac{f_\lambda(s-g(D))}{\sum_{x\in T(D_{-i},\perp)} f_\lambda(s-x)\cdot\Pr[T(D_{-i},\perp)=x]}\right)\right| \\
\leq\ &\ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}}+\beta e^{\frac{(b-a)m}{\lambda}}).
\end{aligned}$$

Thus, for every $W \subseteq \{0,1\}^*$, we have $\left|\ln\left(\frac{\Pr[Out_A(A(z)\leftrightarrow San(D))\in W]}{\Pr[S(z,T(D_{-i},\perp),i,n)\in W]}\right)\right| \leq \ln((1-\beta)e^{\frac{\Delta(g)+\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}})$. □

**Corollary 1.** *Suppose $g : X^n \to [a,b]^m$ has $(\delta,\beta)$-sample complexity with respect to $RS(k(\cdot))$ for some $\beta < \frac{1-k(n)/n}{2}$. Then, the mechanism $San(D) = g(D) + (X_1,\ldots,X_m)$, where $X_j \sim Lap(\lambda)$ for $j = 1,\ldots,m$ independently, is $\ln((1-\beta)e^{\frac{3\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}})$-zero-knowledge private with respect to $RS(k(\cdot))$.*

*Proof.* This follows from combining Proposition 6 and Lemma 1.

Using Proposition 6, we can recover the basic mechanism in [8] that is $\epsilon$-differentially private.

**Corollary 2.** *Let $g : X^n \to [a,b]^m$ and $\epsilon > 0$. A mechanism San for $g$ that adds $Lap(\frac{\Delta(g)}{\epsilon})$ noise to $g(D)$ is $\epsilon$-zero-knowledge private with respect to $RS(n)$.*

*Proof.* We note that every function $g : X^n \to \mathbb{R}^m$ has $(0,0)$-sample complexity with respect to $RS(n)$. The corollary follows by applying Proposition 6.

We now show how the zero-knowledge privacy and utility properties of a mechanism computing a function is related to the sample complexity of the function. A class of algorithms *agg* is said to be *closed under postprocessing* if for any $T \in agg$ and any algorithm $M$, the composition of $M$ and $T$ (i.e., the algorithm that first runs $T$ and then runs $M$ on the output of $T$) is also in *agg*. We note that $RS(k(\cdot))$ is closed under postprocessing.

**Proposition 7.** *Let agg be any class of algorithms that is closed under postprocessing, and suppose a function $g : X^n \to \mathbb{R}^m$ has a mechanism San such that the following hold:*

- *Utility:* $\Pr[||San(D) - g(D)||_1 \leq \delta] \geq 1-\beta$ *for every* $D \in X^n$
- *Privacy: San is $\epsilon$-zero-knowledge private with respect to agg.*

*Then, $g$ has $(\delta, \frac{\beta+(e^\epsilon-1)}{e^\epsilon})$-sample complexity with respect to agg.*

See the full version of this paper for the proof. The intuition is that the zero-knowledge privacy of *San* guarantees that *San* can be simulated by a simulator $S$ that is given aggregate information provided by some algorithm $T \in agg$. Thus, an algorithm that runs $T$ and then $S$ will be able to approximate $g$ with accuracy similar to that of *San*.

### 3.1 Some Simple Examples of Zero-Knowledge Private Mechanisms

*Example 2 (***Averages).** Fix $n > 0$, $k = k(n)$. Let $avg : [0,1]^n \to [0,1]$ be defined by $avg(D) = \frac{\sum_{i=1}^n D_i}{n}$, and let $San(D) = avg(D) + Lap(\lambda)$, where $\lambda > 0$. Let $T$ be an algorithm that, on input a database $D \in [0,1]^n$, chooses $k$ random samples from $D$ (uniformly), and then outputs the average of the $k$ random

samples. By Hoeffding's inequality, we have $\Pr[|T(D) - avg(D)| \leq \delta] \geq 1 - 2e^{-2k\delta^2}$. Thus, $avg$ has $(\delta, 2e^{-2k\delta^2})$-sample complexity with respect to $RS(k(\cdot))$. By Proposition 6, $San$ is $\ln(e^{\frac{1}{\lambda}(\frac{1}{n}+\delta)} + 2e^{\frac{1}{\lambda}-2k\delta^2})$-zero-knowledge private with respect to $RS(k(\cdot))$.

Let $\epsilon \in (0, 1]$. We choose $\delta = \frac{1}{k^{1/3}}$ and $\lambda = \frac{1}{\epsilon}(\frac{1}{n} + \delta) = \frac{1}{\epsilon}(\frac{1}{n} + \frac{1}{k^{1/3}})$ so that $\ln(e^{\frac{1}{\lambda}(\frac{1}{n}+\delta)} + 2e^{\frac{1}{\lambda}-2k\delta^2}) = \ln(e^{\epsilon} + 2e^{\frac{\epsilon}{1/n+k^{-1/3}}-2k^{1/3}}) \leq \ln(e^{\epsilon} + 2e^{-k^{1/3}}) \leq \epsilon + 2e^{-k^{1/3}}$. Thus, we have the following result:

- By adding $Lap(\frac{1}{\epsilon}(\frac{1}{n} + \frac{1}{k^{1/3}})) = Lap(O(\frac{1}{\epsilon k^{1/3}}))$ noise to $avg(D)$, $San$ is $(\epsilon + 2e^{-k^{1/3}})$-zero-knowledge private with respect to $RS(k(\cdot))$.

*Example 3 (***Fraction of rows satisfying some property $P$).** Let $P : X \to \{0, 1\}$ be the predicate representing some property of a row. Let $g : X^n \to [0, 1]$ be defined by $g(D) = \frac{\sum_{i=1}^{n} P(D_i)}{n}$, which is the fraction of rows satisfying property $P$. Since $g(D)$ can be viewed as the average of the numbers $\{P(D_i)\}_{i=1}^{n}$, we can get the same result as in the example for averages.

*Example 4 (***Histograms).** We can easily construct a zero-knowledge private mechanism (with respect to $RS(k(\cdot))$) that computes a histogram with $m$ bins by estimating each bin count separately using $k(n)/m$ random samples each and then applying Proposition 6. Alternatively, we can construct a mechanism by composing $San_i$ for $i = 1, \ldots, m$, where $San_i$ is any zero-knowledge private mechanism (with respect to $RS(\frac{1}{m}k(\cdot))$) for estimating the number of rows in the $i^{\text{th}}$ bin, and then applying our composition result (Propsition 1).

*Example 5 (***Sample and DP-Sanitize).** Our example mechanism for computing averages comes from the general connection between sample complexity and zero-knowledge privacy (Proposition 6), which holds for any model of aggregate information. For computing averages, we can actually construct a mechanism with (usually) better utility by choosing $k(n)$ random samples without replacement from the input database $D \in X^n$ and then running a differentially private mechanism on the chosen samples. It is not hard to show that such a mechanism is zero-knowledge private with respect to $RS(k(\cdot))$. In general, this "sample and DP-sanitize" method works for query functions that can be approximated using random samples (e.g., averages, fractions, and histograms), and allows us to convert differentially private mechanisms to zero-knowledge private mechanisms with respect to $RS(k(\cdot))$. (See the full version of this paper for more details.)

### 3.2 Answering a Class of Queries Simultaneously

In the full version of this paper, we generalize the notion of sample complexity (with respect to $RS(k(\cdot))$ to classes of query functions and show a connection between differential privacy and zero-knowledge privacy for any class of query functions with low sample complexity. In particular, we show that for any class $\mathcal{Q}$ of query functions that can be approximated simultaneously using random

samples, any differentially private mechanism that is useful for $\mathcal{Q}$ can be converted to a zero-knowledge private mechanism that is useful for $\mathcal{Q}$, similar to the "Sample and DP-sanitize" method. We also show that any class of fraction queries (functions that compute the fraction of rows satisfying some property $P$) with low VC dimension can be approximated simultaneously using random samples, so we can use the differentially private mechanisms in [2] and [10] to obtain zero-knowledge private mechanisms (with respect to $RS(k(\cdot))$ for any class of fraction queries with low VC dimension.

## 4 Zero-Knowledge Private Release of Graph Properties

In this section, we first generalize statistical (row) databases to graphs with personal data so that we can model a social network and privately release information that is dependent on the graph structure. We then discuss how to model privacy in a social network, and we construct a sample of zero-knowledge private mechanisms that release certain information about the graph structure of a social network.

We represent a social network using a graph whose vertices correspond to people (or other social entities) and whose edges correspond to social links between them, and a vertex can have certain personal data associated with it. There are various types of information about a social network one may want to release, such as information about the people's data, information about the structure of the social network, and/or information that is dependent on both. In general, we want to ensure privacy of each person's personal data as well as the person's links to other people (i.e., the list of people the person is linked to via edges).

To formally model privacy in social networks, let $\mathcal{G}_n$ be a class of graphs on $n$ vertices where each vertex includes personal data. (When we refer to a graph $G \in \mathcal{G}_n$, the graph always includes the personal data of each vertex.) The graph structure is represented by an adjacency matrix, and each vertex's personal data is represented by a tuple in $X$. For the privacy of individuals, we use our zero-knowledge privacy definition with some minor modifications:

- $\epsilon$-zero-knowledge privacy is defined as before except we change "database $D \in X^n$" to "graph $D \in \mathcal{G}_n$", and we define $(D_{-i}, \perp)$ to be the graph $D$ except the personal data of vertex $i$ is replaced by $\perp$, and all the edges incident to vertex $i$ are removed (by setting the corresponding entries in the adjacency matrix to 0); thus $(D_{-i}, \perp)$ is essentially $D$ with person $i$'s personal data and links removed.

We now consider functions $g : \mathcal{G}_n \to \mathbb{R}^m$, and we redefine the $L_1$-sensitivity of $g$ to be $\Delta(g) = \max\{||g(D') - g(D'')||_1 : D', D'' \in \mathcal{G}_n \text{ s.t. } (D'_{-i}, \perp) = (D''_{-i}, \perp) \text{ for some } i \in [n]\}$. We also redefine $RS(k(\cdot))$ so that the algorithms in $RS(k(\cdot))$ are given a graph $D \in \mathcal{G}_n$ and are allowed to choose $k(n)$ random vertices without replacement and read their personal data; however, the algorithms are not allowed to read the structure of the graph, i.e., the adjacency

matrix. It is easy to verify that all our previous results still hold when we consider functions $g : \mathcal{G}_n \to \mathbb{R}^m$ on graphs and use the new definition of $\Delta(g)$ and $RS(k(\cdot))$.

Since a social network has more structure than a statistical database containing a list of values, we consider more general models of aggregate information that allow us to release more information about social networks:

- $RSE(k(\cdot), s) = k(\cdot)$ random samples with exploration: the class of algorithms $T$ such that on input a graph $D \in \mathcal{G}_n$, $T$ chooses $k(n)$ random vertices uniformly without replacement. For each chosen vertex $v$, $T$ is allowed to explore the graph locally at $v$ until $s$ vertices (including the sampled vertex) have been visited. The data of any visited vertex can be read. (RSE stands for "random samples with exploration".)
- $RSN(k(\cdot), d) = k(\cdot)$ random samples with neighborhood: same as $RSE(k(\cdot), s)$ except that while exploring locally, instead of exploring until $s$ vertices have been visited, $T$ is allowed to explore up to a distance of $d$ from the sampled vertex. (RSN stands for "random samples with neighborhood".)

Note that these models of aggregate information include $RS(k(\cdot))$ as a special case. We can also consider variants of these models where instead of allowing the data of any visited vertex to be read, only the data of the $k(n)$ randomly chosen vertices can be read. (The data of the "explored" vertices cannot be read.)

*Remark 7.* In the above models, vertices (people) in the graph with high degree may be visited with higher probability than those with low degree. Thus, the privacy of these people may be less protected. However, this is often the case in social networks, where people with very many friends will naturally have less privacy than those with few friends.

We now show how to combine Proposition 6 (the connection between sample complexity and zero-knowledge privacy) with recent sublinear time algorithms to privately release information about the graph structure of a social network. For simplicity, we assume that the degree of every vertex is bounded by some constant $d_{\max}$ (which is often the case in a social network anyway).[3]

Let $\mathcal{G}_n$ be the set of all graphs on $n$ vertices where every vertex has degree at most $d_{\max}$. We assume that $d_{\max}$ is publicly known. Let $M = \frac{d_{\max} n}{2}$ be an upper bound on the number of edges of a graph in $\mathcal{G}_n$. For any graph $G \in \mathcal{G}$, the (relative) distance from $G$ to the some property $\Pi$, denoted $dist(G, \Pi)$, is the least number of edges that need to be modified (added/removed) in $G$ in order to make it satisfy property $\Pi$, divided by $M$.

**Theorem 1.** *Let Conn, Eul, and CycF be the property of being connected, Eulerian*[4]*, and cycle-free, respectively. Let $\bar{d}(G)$ denote the average degree of a vertex in $G$. Then, for the class of graphs $\mathcal{G}_n$, we have the following results:*

[3] Weaker results can still be established without this assumption.

[4] A graph $G$ is Eulerian if there exists a path in $G$ that traverses every edge of $G$ exactly once.

1. *The mechanism* $San(G) = dist(G, Conn) + Lap(\frac{2/n+\delta}{\epsilon})$ *is* $\epsilon + e^{-(K-\epsilon/\delta)}$*-zero-knowledge private with respect to* $RSE(k(\cdot), s)$*, where* $k(n) = O(\frac{K}{(\delta d_{\max})^2})$ *and* $s = O(\frac{1}{\delta d_{\max}})$*.*
2. *The mechanism* $San(G) = dist(G, Eul) + Lap(\frac{4/n+\delta}{\epsilon})$ *is* $\epsilon + e^{-(K-\epsilon/\delta)}$*-zero-knowledge private with respect to* $RSE(k(\cdot), s)$*, where* $k(n) = O(\frac{K}{(\delta d_{\max})^2})$ *and* $s = O(\frac{1}{\delta d_{\max}})$*.*
3. *The mechanism* $San(G) = dist(G, CycF) + Lap(\frac{2/n+\delta}{\epsilon})$ *is* $\epsilon + e^{-(K-\epsilon/\delta)}$*-zero-knowledge private with respect to* $RSE(k(\cdot), s)$*, where* $k(n) = O(\frac{K}{\delta^2})$ *and* $s = O(\frac{1}{\delta d_{\max}})$*.*
4. *The mechanism* $San(G) = \bar{d}(G) + Lap(\frac{2d_{\max}/n+\delta L}{\epsilon})$ *is* $\epsilon + e^{-(K-\epsilon/\delta)}$*-zero-knowledge private with respect to* $RSN(k(\cdot), 2)$*, where* $k(n) = O(K\sqrt{n}\log^2 n \cdot \frac{1}{\delta^{9/2}}\log(\frac{1}{\delta}))$*. (Here, we further assume that every graph in* $\mathcal{G}$ *has no isolated vertices and the average degree of a vertex is bounded by* $L$*.)*

The results of the above theorem are obtained by combining Proposition 6 (the connection between sample complexity and zero-knowledge privacy) with sublinear time algorithms from [22] (for results 1, 2, and 3) and [15] (for result 4). Intuitively, the sublinear algorithms give bounds on the sample complexity of the functions ($dist(G, Conn)$, etc.) with respect to $RSE(k(\cdot), s)$ or $RSN(k(\cdot), d)$.

There are already many (non-private) sublinear time algorithms for computing information about graphs whose accuracy is proved formally (e.g., see [15, 3, 22, 13, 18, 14, 24]) or demonstrated empirically (e.g, see [21, 20]). We leave for future work to investigate whether these (or other) sublinear algorithms can be used to get zero-knowledge private mechanisms.

## 5 Acknowledgements

We thank Cynthia Dwork, Ilya Mironov, and Omer Reingold for helpful discussions, and we also thank the anonymous reviewers for their helpful comments.

This material is based upon work supported by the National Science Foundation under Grants 0627680 and 1012593, by the New York State Foundation for Science, Technology, and Innovation under Agreement C050061, and by the iAd Project funded by the Research Council of Norway. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

## References

1. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In: WWW '07: Proc. of the 16th international conference on World Wide Web. pp. 181–190 (2007)
2. Blum, A., Ligett, K., Roth, A.: A learning theory approach to non-interactive database privacy. In: STOC '08: Proc. of the 40th annual ACM symposium on Theory of computing. pp. 609–618 (2008)

3. Chazelle, B., Rubinfeld, R., Trevisan, L.: Approximating the minimum spanning tree weight in sublinear time. SIAM J. Comput. 34(6), 1370–1379 (2005)
4. Chen, B.C., Kifer, D., LeFevre, K., Machanavajjhala, A.: Privacy-preserving data publishing. Foundations and Trends in Databases 2(1-2), 1–167 (2009)
5. Dalenius, T.: Towards a methodology for statistical disclosure control. Statistik Tidskrift 15, 429444 (1977)
6. Dwork, C.: The differential privacy frontier. In: Proc. of the 6th Theory of Cryptography Conference (TCC) (2009)
7. Dwork, C.: Differential privacy. In: ICALP. pp. 1–12 (2006)
8. Dwork, C., Mcsherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Proc. of the 3rd Theory of Cryptography Conference. pp. 265–284 (2006)
9. Dwork, C., Naor, M.: On the difficulties of disclosure prevention in statistical databases or the case for differential privacy (2008)
10. Dwork, C., Rothblum, G., Vadhan, S.: Boosting and differential privacy. In: Proc. of the 51st Annual IEEE Symposium on Foundations of Computer Science (2010)
11. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey of recent developments. ACM Comput. Surv. 42(4), 1–53 (2010)
12. Gehrke, J., Lui, E., Pass, R.: Towards privacy for social networks: A zero-knowledge based definition of privacy (2011), manuscript
13. Goldreich, O., Ron, D.: Property testing in bounded degree graphs. In: Proc. of the 29th annual ACM symposium on Theory of computing. pp. 406–415 (1997)
14. Goldreich, O., Ron, D.: A sublinear bipartiteness tester for bounded degree graphs. In: Proc. of the 30th annual ACM Symposium on Theory of Computing. pp. 289–298 (1998)
15. Goldreich, O., Ron, D.: Approximating average parameters of graphs. Random Struct. Algorithms 32(4), 473–493 (2008)
16. Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P.: Resisting structural re-identification in anonymized social networks. Proc. VLDB Endow. 1, 102–114 (August 2008)
17. Jernigan, C., Mistree, B.: Gaydar. http://www.telegraph.co.uk/technology/facebook/6213590/Gay-men-can-be-identified-by-their-Facebook-friends.html (2009)
18. Kaufman, T., Krivelevich, M., Ron, D.: Tight bounds for testing bipartiteness in general graphs. SIAM J. Comput. 33(6), 1441–1483 (2004)
19. Kifer, D.: Attacks on privacy and definetti's theorem. In: SIGMOD Conference. pp. 127–138 (2009)
20. Krishnamurthy, V., Faloutsos, M., Chrobak, M., Lao, L., Cui, J.H., Percus, A.G.: Reducing large internet topologies for faster simulations. In: IFIP NETWORKING (2005)
21. Leskovec, J., Faloutsos, C.: Sampling from large graphs. In: KDD '06: Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 631–636 (2006)
22. Marko, S., Ron, D.: Approximating the distance to properties in bounded-degree and general sparse graphs. ACM Trans. Algorithms 5(2), 1–28 (2009)
23. Newman, M.E.J.: Ego-centered networks and the ripple effect. Social Networks 25(1), 83 – 95 (2003)
24. Parnas, M., Ron, D.: Testing the diameter of graphs. Random Struct. Algorithms 20(2), 165–183 (2002)