On the Privacy of Social Networks with Personal Privacy Choices

Anonymous Authors
Please do NOT provide authors' names and affiliations
in the paper submitted for review, but keep this placeholder.
ISIT23 follows a **double-blind reviewing policy**.

***Abstract*—We consider the problem of designing privacy mechanisms for social networks' data when users can choose their personal privacy settings. Each user in the graph has personal data and can choose their privacy settings to be: ON or OFF, indicating whether the node requires privacy or not. Moreover, the nodes' data are correlated and form a Markov random field with respect to the graph. We suppose that the users' data is to be shared with a third party, such as electoral or ad campaigns, while respecting the different privacy choices of the users.**

The notion of privacy we use is a variation of differential privacy called dependent differential privacy that is well-equipped to handle the correlated nature of the data. The goal is to preserve the required privacy by releasing a noisy version of each user's data while minimizing the expected error. We focus on the class of one-hop mechanisms in which each node's released data depends on its own and its neighbors' data. This class of mechanisms leads to scalable algorithms that can be implemented on the nodes in parallel. We present OneHop Algorithm and show that it outputs the privatized data while respecting the privacy settings of each user in the presence of correlation. To give more insight, we consider two examples, star and complete graphs, and compare the privacy-utility trade-off of different versions of our algorithm.

I. Introduction

Privacy is now at the forefront of the challenges encountered by online users. A growing trend, which is sometimes imposed by law [1], [2], is to grant individuals more control in deciding the level of privacy they require. This can come under different forms, such as being able to choose privacy settings in websites, mobile apps and social networks. However, due to the correlation in users' online behavior, a user's privacy choices can be misleading in terms of guaranteed privacy. For example, consider the setting of users in a social network. Typically, each user decides his privacy level on his own. However, the privacy choices that his friends make can severely jeopardize his privacy due to the correlation of their data. For instance, a user with a private profile may believe that his personal data is protected. Suppose that, during elections, many of his friends have public profiles and like and retweet left-leaning political opinions. Then, it is safe to guess that this private user will most likely vote left, rendering obsolete his privacy settings.

In general, we can think abstractly of this problem as having a graph in which each node has personal data. Our main motivation is social networks applications, where the graph nodes represent users and the edges or links represent a friendship or "follow" relations[1]. As a first step in modeling the privacy settings, we will assume that each user can choose his privacy to be ON or OFF. Users with privacy OFF are comfortable publicly sharing profile attributes such as gender, political beliefs, likes, dislikes, etc. In contrast, users with privacy ON want to maintain private their personal information. Figure 1 illustrates this setup, in which nodes in a social network can have their privacy being ON or OFF.

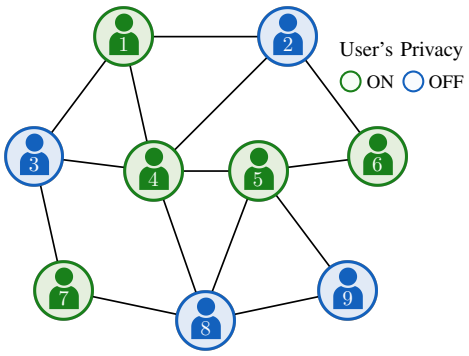


Fig. 1: An example of a social network represented as an ON-OFF privacy graph. The nodes represent users, and the edges represent friendship links. Some users require privacy while others do not, which is indicated by privacy being ON or OFF.

We suppose that the users' data (or a subset of them) is to be shared with a third party, such as electoral or ad campaigns and social or public health studies. To quantify privacy, we will assume that users that are ON all require the same level of differential privacy. Redacting the data before its release by removing the information of ON nodes may not be enough, since correlation can be used to guess the missing data [3], [4]. The main problem becomes to design an algorithm that decides on each user's released data while respecting the users' privacy settings and maximizing the utility of the released data.

A. Related Work

The interplay between privacy and correlation was explored in other ON-OFF privacy settings, including private information retrieval with correlation over time [5], [6], genomic privacy [7] and location privacy [8]. The privacy considered there was information theoretic perfect privacy.

[1]Our model goes beyond social networks and is relevant to any privacy scenario with nodes with local data in a graph, such as IoT networks.

The effect of the social network structure in the context of recommendation systems has been considered in [9], [10], namely how users' similarities, i.e., profile similarity and ratings, correlate to the social relations among them. The importance of correlation in social networks was highlighted in [3], [4], where it was shown that the graph structure and known users' attributes could be used to predict the unknown attributes of other users. Moreover, the work of [11], [12] proposed schemes to protect private information by selectively falsifying or removing some users' traits (data) based on the social graph. However, these methods differ from the work presented in this paper in that they lack rigorous privacy guarantees, such as differential privacy used here.

Differential privacy (DP) [13] is a technique that ensures formal privacy guarantees for algorithms. Previous works on DP involving graphs [14]–[22] focused on graph statistics, such as estimating degree distributions [14], counting subgraphs [15] (e.g., number of triangles), etc. Other related works studied DP algorithms for hiding the presence or absence of an arbitrary set of edges in the graph [18]–[22]. Our paper, however, has a different focus and assumes the graph structure is public and aims at protecting the data carried by nodes with varying privacy settings while accounting for the correlation among their data.

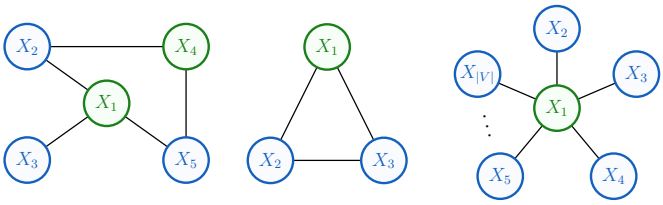
Although DP has been recognized as a powerful notion, it has been observed by [23] that it may not work as expected when the database entries have dependencies. To address differential privacy for correlated data, prior work has explored various privacy metrics, such as pufferfish privacy [24], [25] and dependent differential privacy (DDP) [26]–[28]. Particularly, this paper adopts the DDP definition from [28] to ensure a desired level of privacy regardless of the correlation among the users' data.

B. Contribution

In this paper, we consider the problems of designing privacy mechanisms for social network data when users have different privacy settings. The social network is modeled by a graph. Each node on the graph represents a user that has a binary data, as well as a privacy status that is either ON or OFF. We assume that the users' data are correlated and form a Markov random field with respect to the graph.

We employ the notion of dependent differential privacy (DDP) introduced in [28] to ensure differential privacy in the presence of correlation. The goal is to release noisy versions of the users' data that are as close as possible to their true data while preserving the required privacy.

We focus on the class of One-Hop Mechanisms, in which each node's released data depends on its own and its neighbors data. This class of mechanisms leads to scalable algorithms that can be efficiently run on large-scale networks. We present OneHop Algorithm and show that it outputs the privatized data while respecting the privacy settings of each user in spite the presence of correlation among the users data. To give more insight, we consider two examples, star graphs and complete graphs, and compare the privacy-utility trade-off of different versions of our algorithm.



(a) $|V| = 5$ (b) $|V| = 3$ (c) Star Graph

Fig. 2: Examples of ON-OFF graphs where green nodes have privacy ON while blue nodes have privacy OFF. The nodes' data $X_1, \ldots, X_{|V|}$ are correlated and form a Markov random field with respect to the graph.

C. Notation

For any $n \in \mathbb{N}$, we denote $[n] = \{1, \ldots, n\}$. Given a vector $X = (X_1, \ldots, X_n)$, and a set of indices $A \subseteq [n]$, we adopt the notation $X_A = (X_i \in X : i \in A)$, i.e., truncating coordinates not belonging to the set A. Moreover, two random variables X and Y defined on the same outcome space are equal if their values are the same for every outcome in the space, i.e., $X = Y$. Also, when it is clear from context, we drop the random variable from a probability expression and keep its realization, e.g., for random variables X and Y we have $\Pr(X = x|Y = y) = \Pr(x|y)$.

Furthermore, we define an undirected graph G as an ordered pair $G = (V, E)$ comprising a set of vertices V and a set of edges $E \subseteq \{(u, v) : u, v \in V, u \neq v\}$. Two nodes $u, v \in V$ are neighbors if $(u, v) \in E$, and we denote the set of neighbors of node i by $\mathcal{N}_i := \{j \in V : (i, j) \in E\}$.

II. Setting

We start by describing our setting. We use Markov Random Fields to model the correlation among the users' data and describe our privacy requirement.

A. Correlation via Markov Random Fields

We consider the setting where n users are represented by nodes of a graph $G(V, E)$ with node set $V = [n] := \{1, \ldots, n\}$ and edge set E.[2] Each node $i \in V$ has corresponding data or value represented by a random variable X_i. As a first step in tackling this problem, we focus in this paper on the case where X_i is binary, i.e., $X_i \in \{0, 1\}$.[3]

We assume that the collection of random variables, i.e. $X = (X_1, X_2, \ldots, X_n)$, representing the nodes' data, forms a Markov random field (MRF) [29] with respect to the graph. MRFs have been widely used to model statistical relationships on graphs, e.g., [30]–[32]. For conciseness, we assume that the joint probability distribution of $X_1, \ldots, X_n$ is strictly positive. Below we give a formal definition of an MRF.

[2]We assume the graph is connected, though our results can be easily generalized to an unconnected graph.

[3]In a social network application, binary data can represent a user's "likes" and "follows", group membership, and various personal information. In IoT, it can represent detection of an object or an event.

Definition 1 (Markov Random Field [29])**.** Consider an undirected graph $G = (V, E)$ and a vector of random variables $X = (X_1, X_2, \ldots, X_n)$ where $V = [n]$. We say that X_i forms a Markov random field with respect to G if every random variable in X is conditionally independent of all other variables given its neighbors. That is, for all $i \in V$,

$$\Pr\big(X_i | X_{\mathcal{N}_i}, X_{V \setminus \mathcal{N}_i \cup \{i\}}\big) = \Pr(X_i | X_{\mathcal{N}_i}),$$

where $\mathcal{N}_i$ is the set of node i's neighbors.

For example, the Markov random field represented in the graph in Figure 2a implies that X_1 is independent of X_4, given the data of the subset of its neighbors X_2 and X_5.

B. ON-OFF Privacy Status

We assume that the graph data X is to be shared with a third party, such as advertisement agencies, election campaigns, scientific institutions, etc. This has to be accomplished while respecting the privacy settings of each user/node. In particular, we assume that each node in the graph has the option of having its privacy being set to ON or OFF, indicating whether it wants its personal data to be kept private from a third party.

Let $\Omega \subseteq V$ be the set of nodes with privacy ON. For any node $i \in V$, denote the neighbors of node i that have privacy OFF by $\mathcal{N}_i^{\text{Off}}$, i.e.,

$$\mathcal{N}_i^{\text{Off}} := \{i \in V : (i, j) \in E, j \notin \Omega\}.$$

We assume that the graph structure and the nodes' privacy status are publicly known; only the realizations of the nodes' data are hidden.

C. Privacy

We will use Differential Privacy (DP) [13] as our privacy measure. However, as shown in [23], the standard DP definition does not account for the statistical correlation among the data, which may reveal information about each user. Standard DP is a good fit when the users' data is independent. However, in the presence of correlation, which is a principal aspect of our setting, a user's released data should not only ensure his privacy, but also make sure it does not compromise the privacy of other users with correlated data.

Accordingly, we adopt a variation on the original definition of DP, which is *Dependent Differential Privacy* (DDP) [28], and adapt it to our setting. To that end, we define a randomized mechanism $\mathcal{M}$ as a function with domain consisting of all possible binary data input X, where $X = (X_1, \ldots, X_n) \in \{0, 1\}^n$ is a Markov random field with respect to the graph $G(V, E)$. The privatized data $\mathcal{M}(X)$ will be the data shared with the third party.

Definition 2. [Dependent Differential Privacy] A mechanism $\mathcal{M}$ is ON-OFF ϵ-dependent differentially private (ϵ-DDP) iff for every node i whose privacy is ON, i.e., $i \in \Omega$, and every subset $S \subseteq \text{Range}(\mathcal{M})$, we have

$$\Pr(\mathcal{M}(X) \in S | x_i, x_K) \leq e^{\epsilon} \Pr(\mathcal{M}(X) \in S | x_i', x_K),$$

for all $K \subseteq V \setminus \{i\}$, where all the users' values are binary, i.e., $x_i, x_i' \in \{0, 1\}$ and $x_K \in \{0, 1\}^{|K|}$.

Definition 2 is a slightly modified version of DDP from [28], where we consider the nodes' privacy status ON/OFF. Additionally, Definition 2 differs from the traditional DP definition [13] in that it considers conditioning on all possible subsets of $V \setminus \{i\}$. This is to account for all possible side information that a third party could have, including correlations between nodes and their random variable realizations. Moreover, since the ON-OFF DDP condition is valid for any $K \subseteq V \setminus \{i\}$, it is also valid for $K = V \setminus \{i\}$. Therefore, ON-OFF ϵ-dependent differential privacy implies ON-OFF ϵ-differential privacy.

We will need the notion of max-influence introduced in [25], [28], which quantifies how much a change in a set of variables can impact other variables. The max-influence of a random variable X_i on a set of a random variables X_S conditioned on a set of random variables X_K is defined as

$$\mathcal{I}(X_S \nleftrightsquigarrow X_i | X_K) := \ln \max_{x_S, x_i, x_i', x_K} \frac{\Pr(x_S | x_i, x_K)}{\Pr(x_S | x_i', x_K)}, \qquad (1)$$

where, in our case, all the variables are binary and $x_i \neq x_i'$. We assume that $\mathcal{I}(X_S \nleftrightsquigarrow X_i | X_K) = 0$ if $S = \emptyset$.

D. One-Hop Mechanisms

We are interested in decentralized algorithms, in which each node decides on the privatized shared data based on his personal and his neighbors' data. Decentralized algorithms may not achieve the optimal privacy utility trade-off. However, they are attractive since they can efficiently scale to extremely large graphs, typical of social networks. This is because they do not require a central entity that has access to the entire graph's data. Moreover, decentralized algorithms are appealing for ad-hoc networks, such as in IoT.

Towards that end, we consider the class of randomized mechanisms dubbed One-Hop Mechanisms.

Definition 3 (One-Hop Mechanisms)**.** A one-hop mechanism $\mathcal{M}_i$ at node i is a randomized mechanism that takes as input the data X_i of node i and the data of its neighbors with privacy OFF, that is $X_{\mathcal{N}_i^{\text{Off}}}$. More precisely, $\mathcal{M}_i$ maps X_i and $X_{\mathcal{N}_i^{\text{Off}}}$ to a released value $\overline{X}_i$, i.e.,

$$\mathcal{M}_i : \{0, 1\} \times \{0, 1\}^{|\mathcal{N}_i^{\text{Off}}|} \to \{0, 1\}.$$

As a consequence, the one-hop mechanism $\mathcal{M}$ of the whole data can be factorized as

$$\mathcal{M}(X) = \Big(\mathcal{M}_1(x_1, x_{\mathcal{N}_1^{\text{Off}}}), \ldots, \mathcal{M}_n(x_n, x_{\mathcal{N}_n^{\text{Off}}})\Big).$$

We denote by $\overline{X}$ the output of the one-hop mechanism, i.e.,

$$\overline{X} = \mathcal{M}(X) = \big(\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n\big) \in \{0, 1\}^n.$$

In our definition, a one-hop mechanism at node i does not take as input the true data of its neighbors with privacy ON. This accounts for decentralized network scenarios in which nodes with privacy ON do not trust their neighbors, and circumvents a third party from masquerading as a neighbor to learn the private data of a node. Moreover, the mechanism

is non-interactive[4] in the sense that the mechanism at a certain node does not depend on the released data of other nodes.

E. Utility of a Mechanism

To measure the utility, that is the accuracy of a mechanism, we use the expected Hamming distance between the nodes' true data and their released data. That is, we consider

$$\mathbb{E}\big[d(X,\overline{X})\big],$$

where $d(\cdot)$ is the Hamming distance. Here, the expectation is taken over the randomness of the data, X, and the output of the randomized mechanism, $\overline{X}$.

III. Results Summary

Our main result is the OneHop algorithm detailed in Algorithm 1. It takes among its inputs the nodes' data X and outputs the privatized data $\overline{X}$. OneHop is a decentralized algorithm that belongs to the class of one-hop mechanisms and achieves ON-OFF $\epsilon-$DDP as stated in the next theorem.

Theorem 1 (OneHop Algorithm)**.** *Given an undirected graph $G=(V,E)$, let the nodes' data $X=(X_1,\dots,X_{|V|})$ form a Markov random field with respect to the graph. For a given privacy parameter $\epsilon>0$, distribution of the nodes' data, and pre-specified parameters $\alpha_j=\max_{K\subseteq V\setminus\{i\}}\mathcal{I}(X_{\mathcal{N}_i\setminus K}\nleftrightarrow X_i|X_K)$, $j\in\Omega$, Algorithm 1 is ϵ-DDP.*

We defer the proof of Theorem 1 to Appendix A, and for a computation of the utility of Algorithm 1 see Appendix C-B. We note the following important properties of OneHop.

Property 1. *For each node j with privacy ON, OneHop releases the randomized data $\overline{X}_j$, which depends only on the node's own data X_j and that of its neighbors with privacy OFF, i.e., $X_{\mathcal{N}_j^{\text{Off}}}$. In particular, $\overline{X}_j$ does not depend on the released data of other nodes.*

Property 2. *For nodes with privacy OFF, OneHop releases the true value of their data, that is, $\overline{X}_i=X_i$ for all $i\in V\setminus\Omega$, when ϵ is large, i.e., low privacy requirements.*

Property 1 follows directly from the definition of one-hop mechanisms. It enables the decoupling of the data release mechanism at the nodes, even when there are common neighbors. Technically, this allows us to guarantee an important property (see (7) later), which is an essential part of our privacy proof. As a result, the graph nodes in OneHop can be processed independently in any order or simultaneously in parallel, making the algorithm scalable to large networks.

Complexity of OneHop. It is worth noting that the inputs to OneHop include the parameters α_j for all nodes $j\in\Omega$. Given these parameters, the complexity of Algorithm 1 is $\mathcal{O}(n)$. However, computing α_j depends on the joint distribution of the data and can be a bottleneck. For instance, computing

$$\alpha_j=\max_{K\subseteq V\setminus\{i\}}\mathcal{I}(X_{\mathcal{N}_i\setminus K}\nleftrightarrow X_i|X_K), \tag{2}$$

[4]One could consider interactive mechanisms where nodes can observe other nodes released data, particularly their neighbors. However, we focus on non-interactive mechanisms as a first step in tackling this problem.

Algorithm 1: OneHop

Input: Privacy parameter $\epsilon>0$, the graph $G=(V,E)$, the true data of the nodes $x=(x_1,\dots,x_n)$, parameters α_j and the conditional distribution $\Pr(x_j|x_{\mathcal{N}_j^{\text{Off}}})$ for all $j\in\Omega$.

Output: The released vector $\bar{x}=(\bar{x}_1,\dots\bar{x}_n)$.

if $\epsilon>\max_{j\in\Omega}\alpha_j$ **then**
  foreach user $j\in\Omega$ **do**
    Calculate $\epsilon_j=\epsilon-\alpha_j$ and $c_j=\frac{\Pr\left(X_j=0\middle|x_{\mathcal{N}_j^{\text{Off}}}\right)}{\Pr\left(X_j=1\middle|x_{\mathcal{N}_j^{\text{Off}}}\right)}$.
    If $e^{\epsilon_j}\geq\max\{c_j,1/c_j\}$ **then** $\Pr\left(\overline{X}_j=x_j|x_j,x_{\mathcal{N}_j^{\text{Off}}}\right)=\frac{e^{\epsilon_j}}{1+e^{\epsilon_j}}$.
    else if $c_j>1$ **then** $\bar{x}_j=0$.
    else $\bar{x}_j=1$.
  foreach User $i\notin\Omega$ **do** $\bar{x}_i=x_i$.
else Apply Algorithm 2 with ϵ as input.

for all $j\in\Omega$ by brute force can in general have exponential complexity in the number of nodes n. Although for special Markov random fields, one can derive closed form expressions for the α_j's as we show later in our examples.

To avoid this computational bottleneck, we introduce a helpful corollary, which is immediately established by borrowing from [28] the inequality

$$\max_{K\subseteq V\setminus\{i\}}\mathcal{I}(X_{\mathcal{N}_i\setminus K}\nleftrightarrow X_i|X_K)\leq 4\mathcal{I}(X_{\mathcal{N}_i}\nleftrightarrow X_i).$$

Corollary 1. *For any given privacy parameter $\epsilon>0$, and parameter $\alpha_j=4\mathcal{I}(X_{\mathcal{N}_j}\nleftrightarrow X_j)$ is the max-influence of node j on its neighbors, the one-hop random mechanism of Algorithm 1 is ϵ-DDP.*

The key advantage of the choice of α_j as in Corollary 1 is that the calculation of α_j only requires the probability distribution $\Pr\big(X_{\mathcal{N}_j}|X_j\big)$. This allows α_j to be more efficiently computed, as compared to computing it as in (2).

It should also be noted that the particular choices of α_j can lead to varying privacy and utility guarantees. For a given ON-OFF graph and joint distribution of its the nodes, in Appendix C-A, we show that the utility of OneHop, is strictly increasing in α_j for any $j\in\Omega$. Therefore, the choices of α_j in Theorem 1 lead to better utility, compared to the choices of α_j in Corollary 1. See Figure 3 later for a a concrete example.

AllON Algorithm. To gauge the performance of our scheme, we compare it to the straightforward approach where we assume that all the users have privacy ON. We describe this approach in details in Algorithm 2, which we refer to as AllON. For the sake of completeness, we show the privacy of Algorithm 2 (AllON) in the following theorem.

Theorem 2 (AllON Algorithm)**.** *For a given privacy parameter $\epsilon>0$ and distribution of the nodes' data $X=(X_1,\dots,X_{|V|})$, Algorithm 2 is ϵ-DDP.*

Algorithm 2: AllON

Input: Privacy parameter $\epsilon > 0$, the graph $G = (V, E)$, the true data of the nodes $x = (x_1, \ldots, x_n)$, and the marginal distribution $\Pr(x_i)$ for all $i \in V$.

Output: The released vector $\bar{x} = (\bar{x}_1, \ldots \bar{x}_n)$.

Calculate $\epsilon' = \epsilon/n$.

foreach user $i \in V$ **do**

  if $\Pr(X_i = 1) \leq \Pr(X_i = 0) \leq \frac{1}{1+e^{\epsilon'}}$, **then** $\bar{x}_i = 0$.

  else if $\Pr(X_i = 0) < \Pr(X_i = 1) \leq \frac{1}{1+e^{\epsilon'}}$ **then** $\bar{x}_i = 1$.

  else $\Pr(\overline{X}_i = x_i | x_i) = \frac{e^{\epsilon'}}{1+e^{\epsilon'}}$.

The proof of Theorem 2 is deferred to Appendix B. Moreover, we note that the utility of Algorithm 2 is bounded by

$$\mathbb{E}[d(X, \overline{X})] \leq \frac{n}{1 + e^{\epsilon'}}, \tag{3}$$

where $\epsilon' = \frac{\epsilon}{n}$. The proof will be deferred to Appendix C-B. AllON algorithm is particularly useful for small ϵ. For this reason, the AllON algorithm is incorporated as the last line in Algorithm 1 to handle the small values of ϵ, which helps improve the applicability of our OneHop algorithm.

IV. Special Graphs

In this section, we consider two examples of graphs: the star graphs and the complete graphs. We illustrate the workings of our algorithms and the privacy-utility trade-offs they achieve.

A. Star Graphs

We start by looking at the special case of star graphs, as represented in Figure 2c. Without loss of generality, let node 1 be the center of the star graph, having degree $n - 1$ and all other nodes have degree 1. For any node $i \in V \setminus \{1\}$, assume

$$\Pr(X_i = 0 | X_1 = 0) = \Pr(X_i = 1 | X_1 = 1) = \gamma. \tag{4}$$

We consider the case where node 1 has privacy ON, while all other nodes $V \setminus \{1\}$ have privacy OFF. For this special case, for all $j \in \Omega$, we can explicitly calculate α_j of Theorem 1, as detailed in the following corollary.

Corollary 2 (Star Graphs)**.** *Consider a star graph with n nodes as defined above with data distributed according to* (4)*. The parameters α_j in Theorem 1 are given by:*

$$\alpha_j = (n-1) \ln\left(\max\left\{ \frac{\gamma}{1-\gamma}, \frac{1-\gamma}{\gamma} \right\} \right), \quad j = 1, \ldots, n.$$

And for any privacy parameter $\epsilon > 0$, Algorithm 1 ran on the star graph is ϵ-DDP.

In Figure 3, we consider the star graph with 3 nodes to study privacy vs. utility trade-offs. The utilities of algorithms 1 and 2 are calculated as described in Appendix C. Figure 3 illustrates the fact that OneHop induced by Corollary 1 always performs equivalently or better, in terms of utility for a given ϵ, than OneHop induced by Corollary 2, i.e., Theorem 1. Moreover, Figure 3 also illustrates that for large ϵ, i.e., low privacy requirements, OneHop generally performs better than the straightforward AllON approach of Algorithm 2.

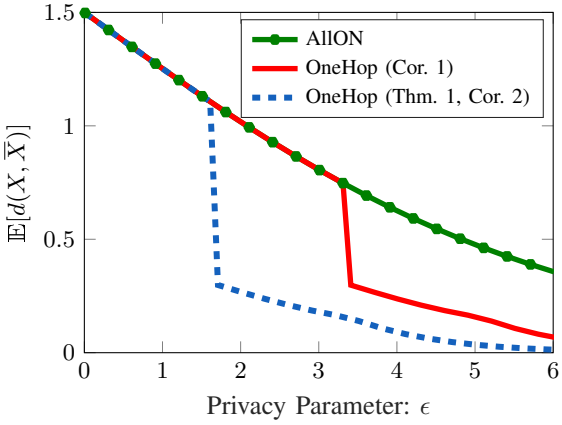


Fig. 3: Privacy vs. utility of our OneHop mechanism described in Algorithm 1, with different input parameters α_j, $j \in \Omega$. Particularly we consider the α_j's induced by corollaries 1 and 2. We also compare OneHop to the straightforward AllON mechanism described in Algorithm 2. These comparison are for the special case of a star graph where node 1 is at the center, $\Omega = \{1\}$, $\gamma = 0.7$, and $\Pr(X_1 = 0) = 0.5$.

B. Complete Graphs

Next we consider the example of a complete graph in which all nodes are inter-connected. We consider the case where only one node has privacy ON, say node 1, e.g., Figure 2b.

We consider the correlation model where the probability that all nodes have the same value is $\beta \geq \frac{1}{2}$. More precisely,

$$\Pr(x) = \begin{cases} \frac{\beta}{2} & \text{for all } x \in \{0\}^n \text{ or } \{1\}^n, \\ \frac{1-\beta}{2^n - 2} & \text{otherwise.} \end{cases} \tag{5}$$

For this case, Theorem 1 can be interpreted as follows.

Corollary 3 (Complete Graphs)**.** *Consider a complete graph with n nodes as defined above with data distributed according to* (5)*. The parameters α_j in Theorem 1 are given by:*

$$\alpha_j = \ln\left(\frac{1}{2} + \frac{\beta(2^{n-1} - 1)}{2(1-\beta)} \right), \quad j = 1, \ldots, n.$$

And for any privacy parameter $\epsilon > 0$, Algorithm 1 ran on the complete graph is ϵ-DDP.

Note that, for a fixed number of users n, the parameter α_j in Corollory 3 is increasing as a function of β, where $0.5 < \beta < 1$. Therefore, when the users behave almost identically, i.e., for large β, OneHop reverts to the AllON approach.

V. Conclusion

We consider the problem of designing privacy mechanisms for social networks (graphs) when users have data and can choose their personal privacy settings: ON or OFF. Moreover, the users' data is correlated and modeled by a Markov random field with respect to the graph. We present a one-hop, scalable, and ϵ-dependent differentially private random mechanism. To analyze its privacy vs. utility trade-offs, we compare it to a straightforward approach where all the nodes are assumed to have privacy ON and explore some special case graphs.

REFERENCES

[1] "Regulation (EU) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (general data protection regulation)," *Official Journal L119*, pp. 1–88, 5 2016.

[2] "The california consumer privacy act of 2018," https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

[3] E. Zheleva and L. Getoor, "To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles," in *Proceedings of the 18th International Conference on World Wide Web*, 2009, p. 531–540.

[4] Y. Ding, S. Yan, Y. Zhang, W. Dai, and L. Dong, "Predicting the attributes of social network users using a graph-based machine learning method," *Computer Communications*, vol. 73, pp. 3–11, 2016.

[5] F. Ye, C. Naim, and S. El Rouayheb, "On-off privacy against correlation over time," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2104–2117, 2021.

[6] ——, "On-off privacy in the presence of correlation," *IEEE Transactions on Information Theory*, vol. 67, no. 11, pp. 7438–7457, 2021.

[7] F. Ye, H. Cho, and S. El Rouayheb, "Mechanisms for hiding sensitive genotypes with information-theoretic privacy," *IEEE Transactions on Information Theory*, vol. 68, no. 6, pp. 4090–4105, 2022.

[8] F. Ye and S. El Rouayheb, "Intermittent private information retrieval with application to location privacy," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 3, pp. 927–939, 2022.

[9] A. Said, E. W. De Luca, and S. Albayrak, "How social relationships affect user similarities," in *Proceedings of the International Conference on Intelligent User Interfaces Workshop on Social Recommender Systems, Hong Kong*, 2010.

[10] P. Bonhard and M. A. Sasse, "'Knowing me, knowing you'—using profiles and social networking to improve recommender systems," *BT Technology Journal*, vol. 24, no. 3, pp. 84–98, 2006.

[11] J. He and W. W. Chu, "Protecting private information in online social networks," in *Intelligence and Security Informatics*, 2008, pp. 249–273.

[12] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "Inferring private information using social network data," in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 1145–1146.

[13] C. Dwork, "Differential Privacy," in *The 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, 2006.

[14] W.-Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 123–138.

[15] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "Differentially private data analysis of social networks via restricted sensitivity," in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 2013, pp. 87–96.

[16] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing graphs with node differential privacy," in *Theory of Cryptography Conference*, 2013, pp. 457–476.

[17] J. Ullman and A. Sealfon, "Efficiently estimating erdos-renyi graphs with node differential privacy," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[18] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, 2007, pp. 75–84.

[19] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *2009 Ninth IEEE International Conference on Data Mining*, 2009, pp. 169–178.

[20] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, "Private analysis of graph structure," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 1146–1157, 2011.

[21] A. Gupta, A. Roth, and J. Ullman, "Iterative constructions and private data release," in *Theory of cryptography conference*, 2012, pp. 339–356.

[22] Q. Xiao, R. Chen, and K.-L. Tan, "Differentially private network data release via structural inference," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 911–920.

[23] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 2011, pp. 193–204.

[24] ——, "A rigorous and customizable framework for privacy," in *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, 2012, pp. 77–88.

[25] S. Song, Y. Wang, and K. Chaudhuri, "Pufferfish privacy mechanisms for correlated data," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1291–1306.

[26] C. Liu, S. Chakraborty, and P. Mittal, "Dependence makes you vulnberable: Differential privacy under dependent tuples." in *Network and Distributed System Security Symposium*, vol. 16, 2016, pp. 21–24.

[27] D. Chakrabarti, J. Gao, A. Saraf, G. Schoenebeck, and F.-Y. Yu, "Optimal local bayesian differential privacy over markov chains." in *AAMAS*, 2022, pp. 1563–1565.

[28] J. Zhao, J. Zhang, and H. V. Poor, "Dependent differential privacy for correlated data," in *2017 IEEE Globecom Workshops (GC Wkshps)*, 2017, pp. 1–7.

[29] R. Kindermann and J. L. Snell, *Markov random fields and their applications*. American Mathematical Society, 1983.

[30] R. P. Kindermann and J. L. Snell, "On the relation between markov random fields and social networks," *Journal of Mathematical Sociology*, vol. 7, no. 1, pp. 1–13, 1980.

[31] O. Frank and D. Strauss, "Markov graphs," *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 832–842, 1986.

[32] S. Wasserman and P. Pattison, "Logit models and logistic regressions for social networks: I. an introduction to markov graphs andp," *Psychometrika*, vol. 61, no. 3, pp. 401–425, 1996.

APPENDIX A
PROOF OF THEOREM 1

We start by introducing several direct consequences to the one-hop mechanism (c.f. Definition 3), which may be frequently invoked in later proofs.

First, notice that the one-hop mechanism requires that each node's released data is only a function of the true data X but not a function of other released data $\overline{X}$. Therefore, for any given non-empty disjoint node subsets, i.e., $A, B \subseteq V$ such that $A \cap B = \emptyset$, we have that $\overline{X}_A$ is independent of $\overline{X}_B$ given the true data $X = \{X_1, \ldots, X_{|V|}\}$, i.e.,

$$\Pr(\bar{x}_A, \bar{x}_B|x) = \Pr(\bar{x}_A|x) \Pr(\bar{x}_B|x). \tag{6}$$

Another helpful consequence to the one-hop mechanism is that $\Pr(\bar{x}|x)$ can be factorized, that is, for a given set $\Omega \subseteq V$, i.e., nodes with privacy ON, we have

$$\Pr(\bar{x}|x) = \Pr\big(\bar{x}_{V\setminus\Omega}|x_{V\setminus\Omega}\big) \prod_{j\in\Omega} \Pr\Big(\bar{x}_j|x_j, x_{\mathcal{N}_j^{\text{Off}}}\Big). \tag{7}$$

To see this, we apply the chain rule on the left-hand side of (7), and we have

$$\Pr(\bar{x}|x) = \Pr(\bar{x}_\Omega|x) \Pr\big(\bar{x}_{V\setminus\Omega}|x, \bar{x}_\Omega\big). \tag{8}$$

Comparing the right-hand sides of (7) and (8), we need to show that

$$\Pr\big(\bar{x}_{V\setminus\Omega}|x, \bar{x}_\Omega\big) = \Pr\big(\bar{x}_{V\setminus\Omega}|x_{V\setminus\Omega}\big), \tag{9}$$

and

$$\Pr(\bar{x}_\Omega|x) = \prod_{j\in\Omega} \Pr\Big(\bar{x}_j|x_j, x_{\mathcal{N}_j^{\text{Off}}}\Big), \tag{10}$$

where both (9) and (10) can be implied by the one-hop mechanism. To make it more clear, we examine the two equations. Firstly, by invoking (6), the left-hand side of (9) can be immediately written as

$$\Pr\big(\bar{x}_{V\setminus\Omega}|x, \bar{x}_\Omega\big) = \Pr\big(\bar{x}_{V\setminus\Omega}|x\big) \tag{11}$$

Also, the one-hop mechanism assumption requires that the released data of a node is only dependent on its own data and that of its neighbors that have privacy OFF but not on the data of its neighbors with privacy ON, i.e., the set Ω. Therefore, we have the following Markov chain $X_\Omega \to X_{V\setminus\Omega} \to \overline{X}_{V\setminus\Omega}$, which implies that

$$\Pr\big(\bar{x}_{V\setminus\Omega}|x\big) = \Pr\big(\bar{x}_{V\setminus\Omega}|x_{V\setminus\Omega}\big).$$

Secondly, (10) also follows from the one-hop mechanism assumption which states that the released data of node j is a function of node j's data and the privacy-OFF neighbors. To be precise, for all $j \in \Omega$, the random variables X and $\overline{X}_j$ form the following Markov chain, $X_{V\setminus(\mathcal{N}_j^{\text{Off}}\cup\{j\})} \to X_{\mathcal{N}_j^{\text{Off}}\cup j} \to \overline{X}_j$, which implies that

$$\Pr(\bar{x}_j|x) = \Pr\Big(\bar{x}_j|x_j, x_{\mathcal{N}_j^{\text{Off}}}\Big). \tag{12}$$

As the above equality holds for all $j \in \Omega$, we can easily obtain that

$$\Pr(\bar{x}_\Omega|x) = \prod_{j\in\Omega} \Pr\Big(\bar{x}_j|x_j, x_{\mathcal{N}_j^{\text{Off}}}\Big)$$

by the chain rule.

We are now ready to prove that the algorithm described in Algorithm 1 is private, i.e., satisfies the ϵ-dependent differential privacy defined in Definition 2.

It should be noted that the privacy needs to hold for two cases, i.e., $\epsilon > \max_{j\in\Omega} \alpha_j$ and $\epsilon \le \max_{j\in\Omega} \alpha_j$. The second case $\epsilon \le \max_{j\in\Omega} \alpha_j$ will be separately proved in Appendix B as an individual proof for the privacy of Algorithm 2.

Therefore, unless otherwise mentioned, the rest of this section is about the proof of privacy for Algorithm 1 for the case $\epsilon > \max_{j\in\Omega} \alpha_j$. The proof is adapted from [28].

For a fixed node $i \in V$ and any subset of nodes $K \subseteq V\setminus\{i\}$, let $\widetilde{K} = V \setminus (K \cup \{i\})$. Then we have

$$\begin{aligned}
&\Pr\big(x_{\widetilde{K}}|x_i, x_K\big) \\
&= \Pr\Big(x_{\widetilde{K}\setminus\mathcal{N}_i}|x_{\widetilde{K}\cap\mathcal{N}_i}, x_i, x_K\Big) \Pr\Big(x_{\widetilde{K}\cap\mathcal{N}_i}|x_i, x_K\Big) \\
&\overset{\text{(a)}}{=} \Pr\Big(x_{\widetilde{K}\setminus\mathcal{N}_i}|x_{\mathcal{N}_i\setminus K}, x_i, x_K\Big) \Pr\big(x_{\mathcal{N}_i\setminus K}|x_i, x_K\big) \\
&\overset{\text{(a)}}{=} \Pr\big(x_{V\setminus(K\cup\mathcal{N}_i\cup i)}|x_{\mathcal{N}_i\cup K}, x_i\big) \Pr\big(x_{\mathcal{N}_i\setminus K}|x_i, x_K\big) \\
&\overset{\text{(b)}}{=} \Pr\big(x_{V\setminus(K\cup\mathcal{N}_i\cup i)}|x_{\mathcal{N}_i\cup K}\big) \Pr\big(x_{\mathcal{N}_i\setminus K}|x_i, x_K\big),
\end{aligned} \tag{13}$$

where (a) follows from the fact $\widetilde{K}\cap\mathcal{N}_i = \mathcal{N}_i\setminus K$ and $\widetilde{K}\setminus\mathcal{N}_i = V \setminus (K \cup \mathcal{N}_i \cup \{i\})$, and (b) follows from the Markovity such that X_i is independent of others given its neighbors $X_{\mathcal{N}_i}$.

By substituting $S = \mathcal{N}_i \setminus K$ in the definition of max influence in (1), we know that

$$\begin{aligned}
&\Pr\big(x_{\mathcal{N}_i\setminus K}|x_i, x_K\big) \le \\
&\qquad e^{\mathcal{I}(X_{\mathcal{N}_i\setminus K} \nleftrightarrow X_i|X_K)} \Pr\big(x_{\mathcal{N}_i\setminus K}|x_i', x_K\big),
\end{aligned} \tag{14}$$

for any $x_i, x_i', x_K, x_{\mathcal{N}_i\setminus K}$.

Since the first term in the right-hand side of (13) is not related to x_i (and x_i'), by combining (13) and (14), we can obtain that

$$\Pr\big(x_{\widetilde{K}}|x_i, x_K\big) \le e^{\mathcal{I}(X_{\mathcal{N}_i\setminus K} \nleftrightarrow X_i|X_K)} \Pr\big(x_{\widetilde{K}}|x_i', x_K\big), \tag{15}$$

for all $x_{\widetilde{K}}, x_K, x_i, x_i'$.

It is easy to verify that the direct release probabilities in Algorithm 1 are judiciously chosen to guarantee that

$$\Pr\Big(\bar{x}_i|x_i, x_{\mathcal{N}_i^{\text{Off}}}\Big) \le e^{\epsilon_i} \Pr\Big(\bar{x}_i|x_i', x_{\mathcal{N}_i^{\text{Off}}}\Big), \tag{16}$$

for any $\bar{x}, x_i, x_i', x_{V\setminus\{i\}}$ and $i \in \Omega$.

By multiplying

$$\Pr\big(\bar{x}_{V\setminus\Omega}|x_{V\setminus\Omega}\big) \prod_{j\in\Omega\setminus\{i\}} \Pr\Big(\bar{x}_j|x_j, x_{\mathcal{N}_j^{\text{Off}}}\Big),$$

which does not contain x_i (and x_i'), on both sides of (16), we can obtain that

$$\Pr\big(\bar{x}|x_i, x_{V\setminus\{i\}}\big) \le e^{\epsilon_i} \Pr\big(\bar{x}|x_i', x_{V\setminus\{i\}}\big) \tag{17}$$

by referring to (7).

For any fixed $i \in \Omega$ and given subset $K \subseteq V \setminus \{i\}$, from (15) and (17), we immediately have

$$\Pr\big(x_{\widetilde{K}}|x_i, x_K\big)\Pr\big(\bar{x}|x_i, x_{V\setminus\{i\}}\big) \leq e^{\epsilon_i + \mathcal{I}(X_{\mathcal{N}_i\setminus K} \leftrightsquigarrow X_i|X_K)}\Pr\big(\bar{x}|x_i', x_{V\setminus\{i\}}\big)\Pr\big(x_{\widetilde{K}}|x_i', x_K\big), \tag{18}$$

for all $x_{V\setminus\{i\}}, x_i, x_i'$. By invoking the marginalization over $x_{\widetilde{K}}$, i.e.,

$$\Pr(\bar{x}|x_i, x_K) = \sum_{x_{\widetilde{K}}} \Pr\big(\bar{x}|x_i, x_{V\setminus\{i\}}\big)\Pr\big(x_{\widetilde{K}}|x_i, x_K\big),$$

we sum (18) over all $x_{\widetilde{K}}$ and we can obtain that

$$\Pr(\bar{x}|x_i, x_K) \leq e^{\epsilon_i + \mathcal{I}(X_{\mathcal{N}_i\setminus K} \leftrightsquigarrow X_i|X_K)}\Pr(\bar{x}|x_i', x_K). \tag{19}$$

It is worth noting that $e^{\epsilon_i + \mathcal{I}(X_{\mathcal{N}_i\setminus K} \leftrightsquigarrow X_i|X_K)}$ is a constant rather than a function of $x_{\widetilde{K}}$.

Recall that $\epsilon_i = \epsilon - \alpha_i$ and $\alpha_i = \max\limits_{K\subseteq V\setminus\{i\}} \mathcal{I}(X_{\mathcal{N}_i\setminus K} \leftrightsquigarrow X_i|X_K)$, and thus we have

$$\epsilon = \epsilon_i + \mathcal{I}(X_{\mathcal{N}_i\setminus K} \leftrightsquigarrow X_i|X_K).$$

Therefore, from (19), we have

$$\Pr(\bar{x}|x_i, x_K) \leq e^{\epsilon}\Pr(\bar{x}|x_i', x_K), \tag{20}$$

for all $i \in \Omega$, $K \subseteq V\setminus\{i\}$, $\bar{x}, x_K, x_i, x_i'$, which completes the proof.

Appendix B
Proof of Theorem 2

For any $j \in \Omega$ and $K \subseteq V \setminus \{j\}$, let $L = K \cup \{j\}$. Given privacy parameter $\epsilon > 0$ let $\epsilon' = \frac{\epsilon}{n}$, where the number of nodes $n = |V|$.

For the case

$$\frac{1}{1+e^{\epsilon'}} < \max\{\Pr(X_i = 0), \Pr(X_i = 1)\},$$

we know that

$$\Pr(\bar{x}_i|x_i) = \frac{1}{1+e^{\epsilon'}} \text{ or } \frac{e^{\epsilon'}}{1+e^{\epsilon'}},$$

which yields that

$$\frac{1}{1+e^{\epsilon'}} \leq \Pr(\bar{x}_i|x_i) \leq \frac{e^{\epsilon'}}{1+e^{\epsilon'}},$$

for any $x_i, \bar{x}_i \in \{0,1\}$. It is worth noting that this also immediately implies

$$\Pr(\bar{x}_i|x_i) \leq e^{\epsilon'}\Pr(\bar{x}_i|x_i'). \tag{21}$$

for all $x_i, x_i', \bar{x}_i \in \{0,1\}$. Therefore,

$$\frac{1}{(1+e^{\epsilon'})^{|S|}} \leq \prod_{i\in S}\Pr(\bar{x}_i|x_i) \leq \frac{e^{|S|\epsilon'}}{(1+e^{\epsilon'})^{|S|}} \tag{22}$$

where $S = V \setminus L$.

For any $x \in \{0,1\}^n$, or in particular $x_L \in \{0,1\}^{|L|}$ and $x_S \in \{0,1\}^{|S|}$, by multiplying (22) by $\Pr(x_S|x_L)$, we have

$$\frac{1}{(1+e^{\epsilon'})^{|S|}}\Pr(x_S|x_L) \leq \Pr(x_S|x_L)\prod_{i\in S}\Pr(\bar{x}_i|x_i) \leq \frac{e^{|S|\epsilon'}}{(1+e^{\epsilon'})^{|S|}}\Pr(x_S|x_L). \tag{23}$$

Summing over all possible $x_S \in \{0,1\}^{|S|}$ and recalling that $L = K \cup \{j\}$, we have

$$\frac{1}{(1+e^{\epsilon'})^{|S|}} \leq \sum_{x_S\in\{0,1\}^{|S|}}\Pr(x_S|x_j, x_K)\prod_{i\in S}\Pr(\bar{x}_i|x_i) \leq \frac{e^{|S|\epsilon'}}{(1+e^{\epsilon'})^{|S|}}. \tag{24}$$

Similarly, we have

$$\frac{1}{(1+e^{\epsilon'})^{|S|}} \leq \sum_{x_S\in\{0,1\}^{|S|}}\Pr\big(x_S|x_j', x_K\big)\prod_{i\in S}\Pr(\bar{x}_i|x_i) \leq \frac{e^{|S|\epsilon'}}{(1+e^{\epsilon'})^{|S|}}, \tag{25}$$

where $x_j' \in \{0,1\}$ and $x_j' \neq x_j$. Note that $j \notin S$ so varying the value of x_j does not affect the term $\prod_{i\in S}\Pr(\bar{x}_i|x_i)$.

By combining (24) and (25), we can obtain that

$$\sum_{x_S\in\{0,1\}^{|S|}}\Pr(x_S|x_j, x_K)\prod_{i\in S}\Pr(\bar{x}_i|x_i) \leq e^{|S|\epsilon'}\sum_{x_S\in\{0,1\}^{|S|}}\Pr\big(x_S|x_j', x_K\big)\prod_{i\in S}\Pr(\bar{x}_i|x_i), \tag{26}$$

for $x_j, x_j' \in \{0,1\}$, where $S = V \setminus L$.

Now, consider

$$\begin{aligned}\Pr(\bar{x}|x_j, x_K) = \Pr(\bar{x}|x_L) &\overset{(a)}{=} \Pr\big(\bar{x}_{V\setminus L}|x_L\big)\prod_{\ell\in L}\Pr(\bar{x}_\ell|x_\ell)\\ &\overset{(b)}{=} \left(\sum_{x_{V\setminus L}}\Pr\big(x_{V\setminus L}|x_L\big)\prod_{i\in V\setminus L}\Pr(\bar{x}_i|x_i)\right)\prod_{\ell\in L}\Pr(\bar{x}_\ell|x_\ell)\\ &= \left(\sum_{x_{V\setminus L}}\Pr\big(x_{V\setminus L}|x_L\big)\prod_{i\in V\setminus L}\Pr(\bar{x}_i|x_i)\right)\Pr(\bar{x}_j|x_j) \qquad (27)\\ &\qquad (28)\end{aligned}$$

where (a) follows because the released data of node i only depends on its own true data in Algorithm 2, and (b) follows from the law of total probability and the same reason as (a).

By examining $\sum_{x_{V\setminus L}}\Pr\big(x_{V\setminus L}|x_L\big)\prod_{i\in V\setminus L}\Pr(\bar{x}_i|x_i)$, we can see that it is exactly the left-hand side of (26), so we have

$$\begin{aligned}&\sum_{x_{V\setminus L}}\Pr\big(x_{V\setminus L}|x_L\big)\prod_{i\in V\setminus L}\Pr(\bar{x}_i|x_i)\\ &= \sum_{x_S}\Pr(x_S|x_j, x_K)\prod_{i\in S}\Pr(\bar{x}_i|x_i)\\ &\leq e^{|S|\epsilon'}\sum_{x_S}\Pr\big(x_S|x_j', x_K\big)\prod_{i\in S}\Pr(\bar{x}_i|x_i).\end{aligned}$$

Also, we know that from (21) that

$$\Pr(\bar{x}_j|x_j) \le e^{\epsilon'} \Pr\big(\bar{x}_j|x'_j\big).$$

By plugging both inequalities in (28), we have

$$\begin{aligned}
&\Pr(\bar{x}|x_j, x_K) \\
&\le e^{(|S|+1)\epsilon'} \left(\sum_{x_S} \Pr\big(x_S|x'_j, x_K\big) \prod_{i\in S} \Pr(\bar{x}_i|x_i) \right) \\
&\qquad\qquad \Pr\big(\bar{x}_j|x'_j\big) \prod_{\ell\in K} \Pr(\bar{x}_\ell|x_\ell) \\
&\le e^{\epsilon} \left(\sum_{x_{V\setminus L}} \Pr\big(x_{V\setminus L}|x'_j, x_K\big) \prod_{i\in V\setminus L} \Pr(\bar{x}_i|x_i) \right) \\
&\qquad\qquad \Pr\big(\bar{x}_j|x'_j\big) \prod_{\ell\in K} \Pr(\bar{x}_\ell|x_\ell) \\
&= e^{\epsilon} \Pr\big(\bar{x}|x'_j, x_K\big),
\end{aligned}$$

where the last step follows because $|S| + 1 = |V \setminus L| + 1 \le n - 1 + 1 = n$ and $\epsilon = n\epsilon'$ by definition.

We have shown that for any $j \in \Omega$, $K \subseteq V \setminus \{j\}$, and $x_j, x'_j \in \{0, 1\}$,

$$\Pr(\bar{x}|x_j, x_K) \le e^{\epsilon} \Pr\big(\bar{x}|x'_j, x_K\big)$$

for the case

$$\frac{1}{1 + e^{\epsilon'}} < \max\{\Pr(X_i = 0), \Pr(X_i = 1)\}.$$

For other cases, i.e., when $\frac{1}{1+e^{\epsilon'}} < \max\{\Pr(X_i = 0), \Pr(X_i = 1)\}$ is not satisfied, we can see from Algorithm 2 that $\bar{X}_i, i \in V$ are released as constants, then the proof follows similarly to above.

Appendix C
Utility of Algorithms 1 and 2

A. Utility of Algorithm 1

For given $\epsilon > 0$ and α_j for all $j \in \Omega$, if $\epsilon \ge \max_{j\in\Omega} \alpha_j$ we have the following utility,

$$\begin{aligned}
&\mathbb{E}[d(X, \overline{X})] = \sum_{i\in V} \mathbb{E}[d(X_i, \overline{X}_i)] \\
&\overset{(a)}{=} \sum_{j\in\Omega} \Pr(\overline{X}_j \ne X_j). \\
&= \sum_{j\in\Omega} \sum_{x_{\mathcal{N}_j^{\text{Off}}}} \Pr(\overline{X}_j \ne X_j, x_{\mathcal{N}_j^{\text{Off}}}) \\
&= \sum_{j\in\Omega} \sum_{x_{\mathcal{N}_j^{\text{Off}}}} \Pr\Big(\overline{X}_j = 1|X_j = 0, x_{\mathcal{N}_j^{\text{Off}}}\Big) \Pr\Big(X_j = 0, x_{\mathcal{N}_j^{\text{Off}}}\Big) \\
&\qquad + \Pr\Big(\overline{X}_j = 0|X_j = 1, x_{\mathcal{N}_j^{\text{Off}}}\Big) \Pr\Big(X_j = 1, x_{\mathcal{N}_j^{\text{Off}}}\Big) \\
&\overset{(b)}{=} \sum_{j\in\Omega, x_{\mathcal{N}_j^{\text{Off}}}} \Pr(X_j = 1, x_{\mathcal{N}_j^{\text{Off}}}) \mathbb{1}[1/c_j < e^{\epsilon_j} < c_j] \\
&\qquad + \Pr(X_j = 0, x_{\mathcal{N}_j^{\text{Off}}}) \mathbb{1}[c_j < e^{\epsilon_j} < 1/c_j] \\
&\qquad + \frac{\Pr(x_{\mathcal{N}_j^{\text{Off}}})}{1 + e^{\epsilon_j}} \mathbb{1}[e^{\epsilon_j} \ge \max\{c_j, 1/c_j\}] \quad (29)
\end{aligned}$$

where $\epsilon_j = \epsilon - \alpha_j$ and the indicator function, denoted by $\mathbb{1}[x > y]$, is defined as

$$\mathbb{1}[x > y] := \begin{cases} 1 & \text{if } x > y \\ 0 & \text{otherwise} \end{cases}.$$

We have that (a) follows from Property 2, and (b) follows from the design of Algorithm 1.

If $0 < \epsilon < \max_{j\in\Omega} \alpha_j$ we have the following utility,

$$\mathbb{E}[d(X, \overline{X})] = \sum_{i\in V} \min\left\{ \frac{1}{1 + e^{\epsilon}}, \Pr(X_i = 0), \Pr(X_i = 1) \right\},$$

which follows directly from the utility proof of Algorithm 2 in Section C-B below.

B. Utility of Algorithm 2

For given $\epsilon > 0$, to find the utility of Algorithm 2, consider the following,

$$\begin{aligned}
&\mathbb{E}[d(X, \overline{X})] = \sum_{i\in V} \Pr\big(X_i \ne \overline{X}_i\big) \\
&= \sum_{i\in V} \Pr\big(\overline{X}_i = 1|X_i = 0\big) \Pr(X_i = 0) + \\
&\qquad\qquad \Pr\big(\overline{X}_i = 0|X_i = 1\big) \Pr(X_i = 1) \\
&\overset{(a)}{=} \sum_{i\in V} \min\left\{ \frac{1}{1 + e^{\epsilon/n}}, \Pr(X_i = 0), \Pr(X_i = 1) \right\} \\
&\le \frac{n}{1 + e^{\epsilon/n}}
\end{aligned}$$

where (a) follows directly from the description of Algorithm 2.

Appendix D
Proof of Corollaries

A. Proof of Corollary 2

Without loss of generality assume $\gamma \ge \frac{1}{2}$. The proof is established by expressing

$$\max_{K\subseteq V\setminus\{1\}} \mathcal{I}(X_S \nleftrightarrow X_1|X_K)$$

by a function of γ explicitly.

By letting $S = \mathcal{N}_1 \setminus K$, we have

$$\begin{aligned}
&\max_{K\subseteq V\setminus\{1\}} \mathcal{I}(X_S \nleftrightarrow X_1|X_K) \\
&= \max_{K\subseteq V\setminus\{1\}} \ln \max_{x_{\mathcal{N}_1\setminus K}, x_1, x'_1, x_K} \frac{\Pr\big(x_{\mathcal{N}_1\setminus K}|x_1, x_K\big)}{\Pr\big(x_{\mathcal{N}_1\setminus K}|x'_1, x_K\big)} \\
&\overset{(a)}{=} \max_{K\subseteq V\setminus\{1\}} \ln \max_{x_{\mathcal{N}_1\setminus K}, x_1, x'_1, x_K} \frac{\prod_{i\in\mathcal{N}_1\setminus K} \Pr(x_i|x_1)}{\prod_{i\in\mathcal{N}_1\setminus K} \Pr(x_i|x'_1)} \\
&\overset{(b)}{=} \max_{K\subseteq V\setminus\{1\}} \ln \frac{\gamma^{|\mathcal{N}_1\setminus K|}}{(1-\gamma)^{|\mathcal{N}_1\setminus K|}} \\
&\overset{(c)}{=} (n-1) \ln \frac{\gamma}{1-\gamma},
\end{aligned}$$

where (a) follows from the Markov random field property where a node is independent of everything else given its neighbors, and in this case the only neighbor of a node $i \in V \setminus \{1\}$ is node 1; (b) follows from (4) and the assumption $\gamma \ge \frac{1}{2}$; and (c) follows because the maximizer is when K is an empty set such that $\max_{K\subseteq V\setminus\{1\}} |\mathcal{N}_1 \setminus K| = n - 1$.

B. Proof of Corollary 3

From (5), we know that for any subset of nodes $S \subseteq V$,

$$\Pr(X_S = \{0\}^{|S|}) = \Pr(X_S = \{1\}^{|S|}) = \frac{\beta}{2} + \frac{(1-\beta)(2^{n-|S|}-1)}{2(2^{n-1}-1)} \quad (30)$$

and

$$\Pr(X_S = x_S) = \frac{2^{n-|S|-1}(1-\beta)}{2^{n-1}-1}, \quad (31)$$

for all $x_S \in \{0,1\}^{|S|} \setminus \big(\{0\}^{|S|} \cup \{1\}^{|S|}\big)$.

Same as what we did in the previous subsection, we now explicitly express

$$\max_{K \subseteq V\setminus\{1\}} \mathcal{I}(X_S \leftsquigarrow X_1 | X_K)$$

in terms of a function of β. By letting $S = \mathcal{N}_1 \setminus K$, we have

$$\begin{aligned}
&\max_{K \subseteq V\setminus\{1\}} \mathcal{I}(X_S \leftsquigarrow X_1 | X_K) \\
&\overset{(a)}{=} \max_{K \subset V\setminus\{1\}} \ln \max_{x_{\mathcal{N}_1\setminus K}, x_1, x_1', x_K} \frac{\Pr\big(x_{\mathcal{N}_1\setminus K} | x_1, x_K\big)}{\Pr\big(x_{\mathcal{N}_1\setminus K} | x_1', x_K\big)} \\
&\overset{(b)}{=} \max_{K \subset V\setminus\{1\}} \ln \max_{x_{V\setminus\{1\}}, x_1, x_1'} \frac{\Pr\big(x_{V\setminus\{1\}} | x_1\big)\Pr(x_K | x_1')}{\Pr\big(x_{V\setminus\{1\}} | x_1'\big)\Pr(x_K | x_1)} \\
&= \max_{K \subset V\setminus\{1\}} \ln \max_{x_{V\setminus\{1\}}, x_1, x_1'} \frac{\Pr\big(x_{V\setminus\{1\}}, x_1\big)\Pr(x_K, x_1')}{\Pr\big(x_{V\setminus\{1\}}, x_1'\big)\Pr(x_K, x_1)} \\
&= \ln \max_{K \subset V\setminus\{1\}} \max_{x_{V\setminus\{1\}}, x_1, x_1'} \frac{\Pr\big(x_{V\setminus\{1\}}, x_1\big)\Pr(x_K, x_1')}{\Pr\big(x_{V\setminus\{1\}}, x_1'\big)\Pr(x_K, x_1)},
\end{aligned} \quad (32)$$

where (a) follows from the fact $\mathcal{I}(X_S \leftsquigarrow X_1|X_K) \geq 0$ and the assumption $\mathcal{I}(X_S \leftsquigarrow X_1|X_K) = 0$ when $S = \emptyset$ and (b) follows from $\mathcal{N}_1 = V \setminus \{1\}$ for a complete graph.

Let

$$r(z) = \frac{\beta(2^{n-1}-1) + (1-\beta)(2^{n-z}-1)}{(1-\beta)2^{n-z}},$$

i.e., the right-hand side of (30) over the right-hand side of (31) by substituting $|S| = z$. It is easy to see that $r(z) \geq 1$ and $r(z)$ is an increasing function with respect to z when $n \geq 2$ and $\beta \geq \frac{1}{2}$. Then the maximization in (32) can be written as

$$\max_{k=|K|=0,\ldots,n-2} \max\left\{\frac{r(n)}{r(k+1)}, r(k+1)\right\}$$

by considering all configurations of x_1, x_1', x_k and $x_{V\setminus\{1\}}$.

Since $r(z)$ is increasing, we know that the maximization can only be attained by

$$\begin{aligned}
&\max_{k=|K|=0,\ldots,n-2} \max\left\{\frac{r(n)}{r(k+1)}, r(k+1)\right\} \\
&\qquad = \max\left\{\frac{r(n)}{r(1)}, r(n-1)\right\}.
\end{aligned}$$

Since

$$\begin{aligned}
r(n) &= \frac{\beta\big(2^{n-1}-1\big)}{1-\beta}, \quad r(1) = \frac{2^{n-1}-1}{(1-\beta)2^{n-1}}, \\
r(n-1) &= \frac{\beta(2^{n-1}-1) + (1-\beta)}{2(1-\beta)},
\end{aligned}$$

we can verify that

$$\begin{aligned}
r(n-1) - \frac{r(n)}{r(1)} &= \frac{\beta(2^{n-1}-1) + (1-\beta)}{2(1-\beta)} - \beta 2^{n-1} \\
&\geq 0
\end{aligned}$$

by invoking that $\beta \geq \frac{1}{2}$ and $n \geq 2$.

Therefore, we have

$$\begin{aligned}
&\max_{k=|K|=0,\ldots,n-2} \max\left\{\frac{r(n)}{r(k+1)}, r(k+1)\right\} \\
&\qquad = \frac{\beta(2^{n-1}-1) + (1-\beta)}{2(1-\beta)},
\end{aligned}$$

which implies that

$$\max_{K \subseteq V\setminus\{1\}} \mathcal{I}(X_S \leftsquigarrow X_1|X_K) = \ln\left(\frac{1}{2} + \frac{\beta(2^{n-1}-1)}{2(1-\beta)}\right),$$

and completes the proof.