Towards Plausible Graph Anonymization

Yang Zhang*, Mathias Humbert†, Bartlomiej Surma*,
Praveen Manoharan*, Jilles Vreeken*, Michael Backes*
*CISPA Helmholtz Center for Information Security,
{zhang, bartlomiej.surma, praveen.manoharan, vreeken, backes}@cispa.saarland
†Cyber-Defence Campus, armasuisse Science and Technology, mathias.humbert@armasuisse.ch

Abstract—Social graphs derived from online social interactions contain a wealth of information that is nowadays extensively used by both industry and academia. However, as social graphs contain sensitive information, they need to be properly anonymized before release. Most of the existing graph anonymization mechanisms rely on the perturbation of the original graph's edge set. In this paper, we identify a fundamental weakness of these mechanisms: They neglect the strong structural proximity between friends in social graphs, thus add implausible fake edges for anonymization.

To exploit this weakness, we first propose a metric to quantify an edge's plausibility by relying on graph embedding. Extensive experiments on three real-life social network datasets demonstrate that our plausibility metric can very effectively differentiate fake edges from original edges with AUC (area under the ROC curve) values above 0.95 in most of the cases. We then rely on a Gaussian mixture model to automatically derive the threshold on the edge plausibility values to determine whether an edge is fake, which enables us to recover to a large extent the original graph from the anonymized graph. We further demonstrate that our graph recovery attack jeopardizes the privacy guarantees provided by the considered graph anonymization mechanisms.

To mitigate this vulnerability, we propose a method to generate fake yet plausible edges given the graph structure and incorporate it into the existing anonymization mechanisms. Our evaluation demonstrates that the enhanced mechanisms decrease the chances of graph recovery, reduce the success of graph de-anonymization (up to 30%), and provide even better utility than the existing anonymization mechanisms.

I. INTRODUCTION

The rapid development of online social networks (OSNs) has resulted in an unprecedented scale of social graph data available. Access to such data is invaluable for both the industrial and academic domains. For instance, Amazon or Netflix have leveraged graph data to improve their recommendation services. Moreover, researchers have been using graph data to gain a deeper understanding of many fundamental societal questions, such as people's communication patterns [34], [50], geographical movement [10], [51], and information propagation [23], [39]. These examples demonstrate that the sharing of large-scale graph data can bring significant benefits to the society.

On the downside, graph data also inherently contains very sensitive information about individuals [7], such as their social relations [4], and it can be used to infer private attributes [22]. In order to mitigate privacy risks, it is crucial to properly anonymize the graph data before releasing it to third parties. The naive approach of replacing real identifiers by random numbers has been proven ineffective by Backstrom et al. about a decade ago already [5]. From then on, the research community has been working on developing more robust graph anonymization mechanisms [26], [40], [31], [49], [20]. The majority of the proposed mechanisms focus on perturbing the original edge set of the graph (instead of perturbing the node set) by adding *fake edges* between users, such that the *perturbed graph* satisfies well-established privacy guarantees, such as k-anonymity [44] and differential privacy [11].

A. Contributions

In this paper, we identify a fundamental weakness of the most prominent graph anonymization mechanisms: When creating fake edges, they do not take into account key characteristics of the underlying graph structure, in particular, the higher structural proximity between friends [25], which results in fake edges not being plausible enough compared to the original ones. To exploit this weakness, we first assess the plausibility of each edge by relying on graph embedding [35], [14]. We show that this approach can very effectively detect fake edges (see Figure 1a for an example of the edge plausibility distribution of fake and original edges on a real-life social network dataset), and thus can eventually help recover the original graph to a large extent. We then demonstrate that our graph recovery attack jeopardizes the anonymization mechanisms' privacy guarantees. Finally, we develop enhanced versions of the existing graph anonymization mechanisms that: (i) create plausible edges (Figure 1b), (ii) reduce the risk of graph recovery and graph de-anonymization, (iii) preserve the initial privacy criteria provided by the mechanisms, and (iv) provide even better graph utility (with respect to how well the anonymized graph preserves the structural properties of the original graph).

To illustrate the wide applicability of our approach, we concentrate on two of the best established graph anonymization mechanisms, namely k-DA [26] and SalaDP [40], which provide k-anonymity and differential privacy guarantees, respectively. The reason we choose k-DA and SalaDP is that they are the best graph anonymization schemes with respect to utility and resistance to de-anonymization (in addition to being the most cited). This conclusion is drawn from the evaluation performed by Ji et al. [18].
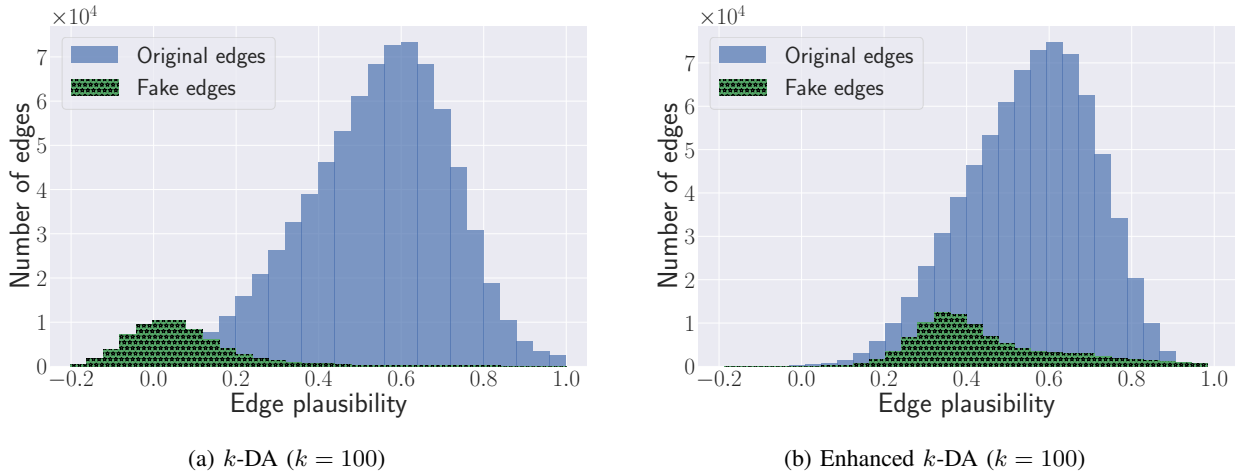
(a) k-DA ($k = 100$)

(b) Enhanced k-DA ($k = 100$)

Fig. 1: Plausibility distributions of fake and original edges in the NO dataset anonymized by (a) the original k-DA and (b) by our enhanced k-DA mechanisms. The edge plausibility is defined in Section III. The NO dataset is collected by Viswanath et al. [47], and k-DA [26] is one of the anonymization mechanisms we concentrate on in this paper.

In the following, we provide an overview of our contributions in this paper.

Edge Plausibility: We measure the plausibility of an edge as the structural proximity between the two users it connects. In the field of link prediction [25], structural proximity is normally measured by human-designed metrics, which only capture partial information of the proximity. Instead, we rely on graph embedding [35], [14] to map users in the anonymized graph into a continuous vector space, where each user's vector comprehensively reflects her structural properties in the graph. Then, we define each edge's plausibility as the similarity between the vectors of the two users this edge connects, and postulate that lower similarity implies lower edge plausibility.

Graph Recovery: We show the effectiveness of our approach in differentiating fake edges from original ones without determining a priori a specific decision threshold on the plausibility metric. For this case, we adopt the AUC (area under the ROC curve) value as the evaluation metric. Extensive experiments performed on three real-life social network datasets show that our plausibility metric achieves excellent performance (corresponding to AUC values greater than 0.95) in most of the cases. Then, observing that the fake and real edges' empirical plausibility follow different Gaussian distributions, we rely on a Gaussian mixture model and maximum a posteriori probability estimate to automatically determine the threshold on the edge plausibility values to detect fake edges. Our experimental results show that this approach achieves strong performance with F1 scores above 0.8 in multiple cases. After deleting the fake edges, we are able to recover, to a large extent, the original graph from the anonymized one.

Privacy Damage: The two anonymization mechanisms we consider follow different threat models and privacy definitions. To precisely quantify the privacy impact of our graph recovery, we propose privacy loss measures tailored to each mechanism we target. As the first anonymization mechanism

assumes the adversary uses the users' degrees to conduct her attack, we evaluate the corresponding privacy impact as the difference between users' degrees in the original, anonymized, and recovered graphs. For the differential privacy mechanism, we measure the magnitude and entropy of noise added to the statistical measurements of the graph. Our experimental results show that the privacy provided by both mechanisms significantly decreases, which demonstrates the vulnerabilities of existing graph anonymization techniques.

Enhancing Graph Anonymization: In order to improve the privacy situation, we propose a method that generates plausible edges while preserving the original privacy guarantees of each mechanism. We rely on statistical sampling to select potential fake edges that follow a similar plausibility distribution as the edges in the original graph. Our experimental results show that our enhanced anonymization mechanisms are less prone to graph recovery (AUC dropping by up to 35%) and preserve higher graph utility compared to the existing anonymization mechanisms. More importantly, we show that our enhanced mechanisms reduce the state-of-the-art graph de-anonymization [32] attack's performance significantly (up to 30% decrease in the number of de-anonymized users).

In summary, we make the following contributions in this paper:

- We perform a graph recovery attack on anonymized social graphs based on graph embedding that captures the structural proximity between users and thus unveils fake edges (i.e., relations) between them.

- We show through extensive experimental evaluation that our graph recovery attack jeopardizes the privacy guarantees provided in two prominent graph anonymization mechanisms.

- We propose enhanced versions of these graph anonymization mechanisms that improve both their privacy and utility provisions.

TABLE I: Notations.

Notation	Description
$\mathcal{G} = (\mathcal{U}, \mathcal{E})$	Social graph
$u \in \mathcal{U}$	A social network user
$\{u, u'\} \in \mathcal{E}$	An edge connecting users u and u'
$\mathcal{A}$	Anonymization mechanism
$\mathcal{G}_\mathcal{A}$	Anonymized social graph
$\kappa(u)$	Friends of user u
$f(u)$	Embedding vector of user u
$s_\mathcal{A}(u, u')$	Plausibility of edge $\{u, u'\}$ in $\mathcal{G}_\mathcal{A}$
$\mathcal{G}_\mathcal{R}$	Recovered social graph
$D(\mathcal{G})$	dK-2 series of $\mathcal{G}$
$\mathcal{G}_\mathcal{F}$	Anonymized graph by enhanced mechanism

B. Organization

The rest of the paper is organized as follows. We introduce the notations, anonymization mechanisms, and threat model used throughout the paper in Section II. Section III presents our edge plausibility definition and Section IV evaluates its effectiveness. The privacy impact of our graph recovery is studied in Section V. In Section VI, we introduce our enhanced graph anonymization mechanisms. Section VII discusses the related work in the field and Section VIII concludes the paper.

II. PRELIMINARIES

In this section, we first introduce the notations, second, describe the two anonymization mechanisms we study, and third, present the threat model.

A. Notations

We model a social graph as an undirected graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where set $\mathcal{U}$ contains the users (nodes) and set $\mathcal{E} \subseteq \{\{u, u'\} | u, u' \in \mathcal{U} \wedge u \neq u'\}$ represents all the edges of the graph. We define by $\mathcal{A}$ the anonymization mechanism which transforms $\mathcal{G}$ to an anonymized graph $\mathcal{G}_\mathcal{A} = (\mathcal{U}, \mathcal{E}_\mathcal{A})$ following the privacy criteria of $\mathcal{A}$. By this definition, we only consider graph anonymization mechanisms that do not add new nodes but only modify edges. This is in line with most of the previous works [26], [52], [40], [31], [49]. We further use $\kappa(u)$ to represent u's friends in $\mathcal{G}$, i.e., $\kappa(u) = \{u' | \{u, u'\} \in \mathcal{E}\}$. Accordingly, $\kappa_\mathcal{A}(u)$ represents u's friends in $\mathcal{G}_\mathcal{A}$. For presentation purposes, we summarize the notations introduced here and in the following sections in Table I.

B. Graph Anonymization Mechanisms

We briefly review the two graph anonymization mechanisms we study. For more details, we refer the readers to the original papers. Note that, to fully understand these two mechanisms, we have also inspected the source code of SecGraph [18], a state-of-the-art software system for evaluating graph anonymization which includes an implementation of both k-DA and SalaDP.

k**-DA [26]:** k-DA follows the notion of k-anonymity in database privacy. The mechanism assumes that the adversary has prior knowledge of its target users' degrees in a social graph, i.e., numbers of friends, and uses this knowledge to identify the targets from the graph. To mitigate this privacy risk, k-DA modifies the original social graph, such that in the resulting anonymized graph, each user shares the same degree with at least $k - 1$ other users.

k-DA takes two steps to achieve its goal. First, it utilizes dynamic programming to construct a k-anonymous degree sequence. Second, the mechanism adds edges[1] to the original graph in order to realize the k-anonymous degree sequence. By calculating the differences between the original degree sequence and the k-anonymous degree sequence, k-DA maintains a list that stores the number of edges needed for each user, namely the user's *residual degree*. When adding an edge for a certain user, k-DA picks the new adjacent user with the highest residual degree.

SalaDP [40]: SalaDP is one of the first and most widely known mechanisms applying differential privacy for graph anonymization. The statistical metric SalaDP concentrates on is the dK-2 series of a graph $\mathcal{G}$ which counts, for each pair (i, j) of node degrees i and j, the number of edges in $\mathcal{G}$ that connect nodes of these degrees. A formal definition of dK-2 series will be provided in Section V.

SalaDP also takes a two-step approach to anonymize a graph. First, the mechanism adds Laplace noise to each element in the original dK-2 series, and obtains a differentially private dK-2 series. Then, it generates the anonymized graph following the new dK-2 series. By checking SecGraph's source code, we find that SalaDP generates the anonymized graph by (mainly) adding fake edges to the original graph in a random manner.[2]

From the above descriptions, we can see that neither of the anonymization mechanisms consider friends' strong structural proximity when adding fake edges. The main hypothesis we investigate is that we can effectively differentiate the fake edges added by such mechanisms from the original edges, using a suitable measure for edge plausibility. We focus on fake added edges (and not on deleted edges) since most of the graph anonymization mechanisms mainly add edges to the original social graph for preserving better graph utility. It is worth noting that our approach (Section III) can also help recover deleted edges on anonymized graphs. However, the underlying search space is then $\mathcal{O}(|\mathcal{U}|^2)$, which is computationally very expensive on large graphs. In the future, we plan to tackle this problem by designing heuristics to efficiently recover the deleted edges.

C. Threat Model

The adversary's goal is to detect fake edges in $\mathcal{G}_\mathcal{A}$, partially recover the original graph, and eventually carry out privacy attacks on the recovered graph. To perform graph recovery, we assume that the adversary only has access to the anonymized graph $\mathcal{G}_\mathcal{A}$ and is aware of the underlying anonymization algorithm. This means that the adversary does not need any information about the original graph $\mathcal{G}$, such as $\mathcal{G}$'s graph structure or any statistical background knowledge related to this graph. Figure 2 depicts a schematic overview of the attack.

---

[1]In its relaxed version, k-DA also deletes a small fraction of edges, but its major operation is still adding edges.

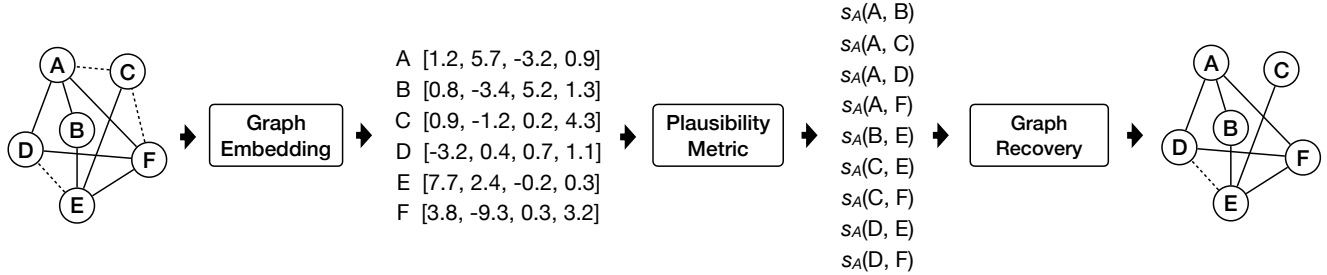[2]Line 252 of `SalaDP.java` in `src/anonymize/` of SecGraph.

Fig. 2: A schematic view of our graph recovery attack. Dashed lines in the graph represent fake added edges; $s_\mathcal{A}(A, B)$ represents the plausibility of edge $\{A, B\}$ (Section III); A [1.2, 5.7, -3.2, 0.9] represents an example embedding vector of A (Section III).

Our graph recovery attack could also be carried out by a service provider (e.g., OSN operator) to check whether there are potential flaws in its anonymized graph data before release.

III. EDGE PLAUSIBILITY

To verify our hypothesis that an edge is fake if the users it connects are structurally distant, we first need to quantify two users' structural proximity in a social graph. Previous work on link prediction provides numerous proximity metrics [25], [6], [2]. However, these metrics are manually designed and only capture partial information of structural proximity. The recent advancement of graph embedding provides us with an alternative approach [35], [45], [14], [4], [16], [38]. In this context, users in a social network are embedded into a continuous vector space, such that each user's vector comprehensively reflects her structural property in the network. Then, for an edge in the anonymized graph, we can define its two users' structural proximity as the similarity of their vectors, and use this similarity as the edge's plausibility.

In this section, we first recall the methodology of graph embedding, and then formally define edge plausibility.

A. Graph Embedding

Graph embedding aims to learn a map f from users in $\mathcal{G}_\mathcal{A}$ to a continuous vector space, i.e.,

$$f : \mathcal{U} \to \mathbb{R}^d,$$

where d, as a hyperparameter, is the dimension of each user's vector. We adopt the state-of-the-art optimization framework, namely Skip-gram [29], [30], to learn f; the corresponding objective function is defined as:

$$\arg \max_f \prod_{u \in \mathcal{U}} \prod_{u' \in \omega(u)} P(u'|f(u)). \qquad (1)$$

Here, the conditional probability $P(u'|f(u))$ is modeled with a softmax function

$$P(u'|f(u)) = \frac{\exp(f(u') \cdot f(u))}{\sum_{v \in \mathcal{U}} \exp(f(v) \cdot f(u))},$$

where $f(u') \cdot f(u)$ is the dot product of the two vectors, and $\omega(u)$ represents u's *neighborhood* in $\mathcal{G}_\mathcal{A}$. To define $\omega(u)$, we use a random walk approach following previous works [35],

[14]. Concretely, we start a random walk from each user in $\mathcal{G}_\mathcal{A}$ for a fixed number of times t, referred to as the *walk times*. Each random walk takes l steps, referred to as the *walk length*. The procedure results in a set of truncated random walk traces, and each user's neighborhood includes the users that appear before and after her[3] in all these random walk traces. Similar to the vector dimension (d), walk length and walk times (l and t) are also hyperparameters. We will choose their values experimentally.

Objective function (Equation 1) implies that if two users share similar neighborhoods in $\mathcal{G}_\mathcal{A}$, then their learned vectors will be closer than those with different neighborhoods. This results in each user's vector being able to preserve her neighborhood and to eventually reflect her structural property in $\mathcal{G}_\mathcal{A}$. To optimize Equation 1, we rely on stochastic gradient descent (SGD) with negative sampling [30]. We omit the details here due to space limitation.

B. Quantifying Edge Plausibility

Given the vectors learned from graph embedding, we define an edge plausibility as the cosine similarity between its two users' vectors. Formally, for $\{u, u'\} \in \mathcal{E}_\mathcal{A}$, its plausibility is defined as:

$$s_\mathcal{A}(u, u') = \frac{f(u) \cdot f(u')}{||f(u)||_2 \, ||f(u')||_2}$$

where $||\cdot||_2$ denotes the L_2-norm. Consequently, if the vectors of two users have higher (cosine) similarity, then the edge connecting these users is more plausible. It is worth noting that as $f(u) \in \mathbb{R}^d$, the range of $s_\mathcal{A}(u, u')$ lies in [-1, 1] instead of [0, 1].

IV. GRAPH RECOVERY

In this section, we first evaluate the effectiveness of our edge plausibility metric on differentiating fake edges from original ones without determining a decision threshold on the edge plausibility a priori. Then, we present a method to automatically decide whether an edge is fake, which allows us to eventually recover the original graph.
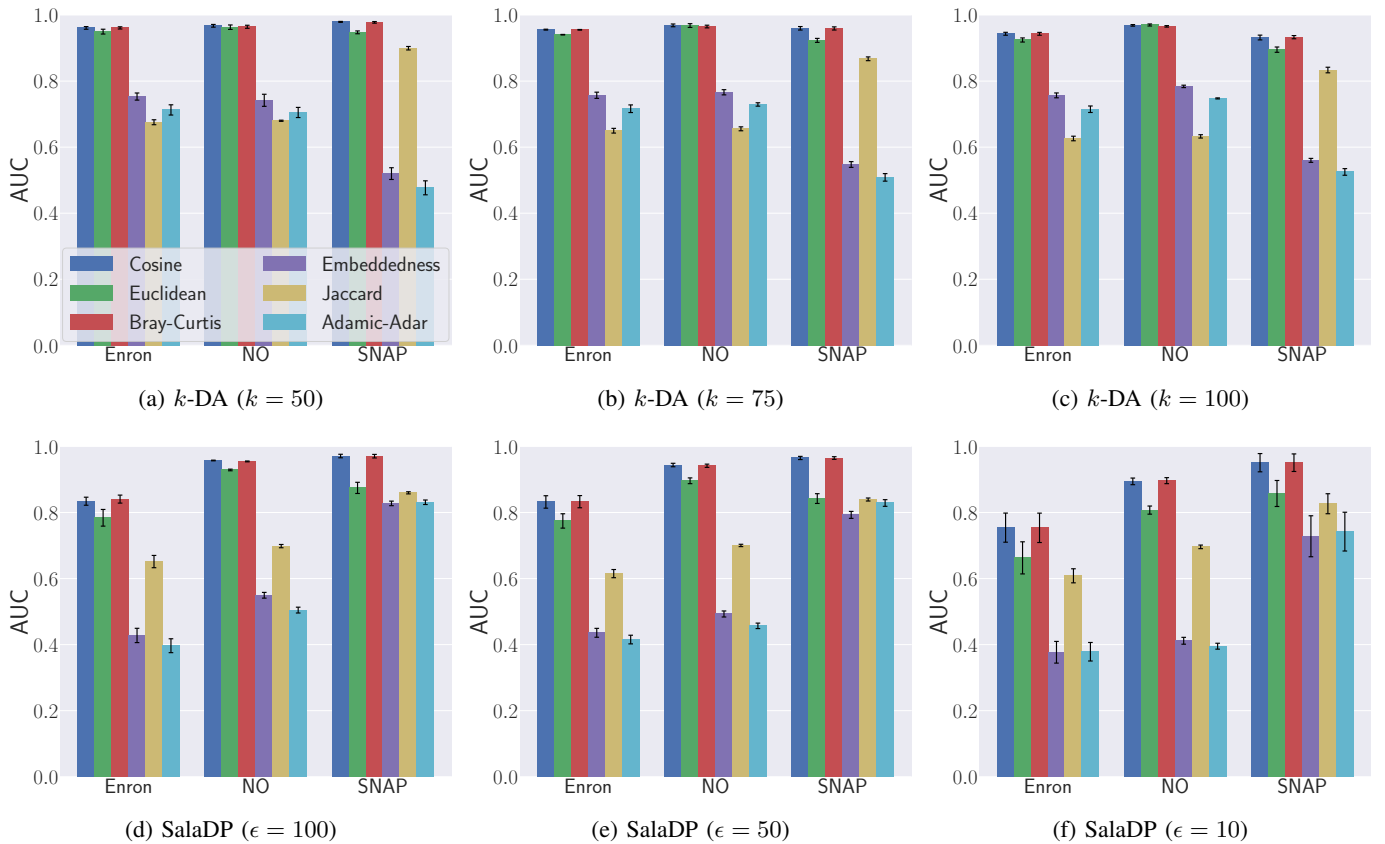
(a) k-DA ($k = 50$) (b) k-DA ($k = 75$) (c) k-DA ($k = 100$)

(d) SalaDP ($\epsilon = 100$) (e) SalaDP ($\epsilon = 50$) (f) SalaDP ($\epsilon = 10$)

Fig. 3: [Higher is better] AUC scores for detecting fake edges for different datasets, structural proximity, distance metrics, and anonymity levels (k resp. ϵ). The embedding approach clearly outperforms all three traditional structural proximity metrics. Moreover, cosine similarity performs best, only matched by Bray-Curtis distance.

TABLE II: Statistics of the datasets.

	Enron	NO	SNAP
Number of users	36,692	63,731	4,039
Number of edges	183,831	817,090	88,234
Average degree	10.020	25.642	43.691
Average clustering coefficient	0.497	0.221	0.606
Number of triangles	727,044	3,501,542	1,612,010

A. Experimental Setup

Datasets: We utilize three datasets for our experiments. The first one, referred to as Enron, is a network of Email communications in the Enron corporation.[4] The second dataset (NO) is collected from Facebook users in the New Orleans area by Viswanath et al. [47]. The third dataset (SNAP) by McAuley and Leskovec is obtained through a survey study [27]. Note that Enron and NO are the two datasets used in the evaluation of SecGraph as well [18]. Table II presents some basic statistics of the three datasets.

Baseline Models and Evaluation Metrics: To demonstrate

---

[3]We select 10 users before and after the considered user following previous works [35], [14], [4].

[4]https://snap.stanford.edu/data/email-Enron.html

the effectiveness of our plausibility metric, which is essentially a structural proximity metric, we compare it with three classical structural proximity metrics, namely, embeddedness (number of common friends), Jaccard index, and Adamic-Adar score [1]. Their formal definition is as the following.

$$\text{Embeddedness} : |\kappa_{\mathcal{A}}(u) \cap \kappa_{\mathcal{A}}(u')|$$

$$\text{Jaccard index} : \frac{|\kappa_{\mathcal{A}}(u) \cap \kappa_{\mathcal{A}}(u')|}{|\kappa_{\mathcal{A}}(u) \cup \kappa_{\mathcal{A}}(u')|}$$

$$\text{Adamic-Adar score} : \sum_{v \in \kappa_{\mathcal{A}}(u) \cap \kappa_{\mathcal{A}}(u')} \frac{1}{\log |\kappa_{\mathcal{A}}(v)|}$$

Recall that cosine similarity is adopted for measuring edge plausibility based on the users' vectors learned from graph embedding. We also test two other vector similarity (distance) metrics, namely the Euclidean distance and the Bray-Curtis distance, defined as follows:

$$\text{Euclidean} : ||f(u) - f(u')||_2$$

$$\text{Bray-Curtis} : \frac{\sum_{i=1}^{d} |f(u)_i - f(u')_i|}{\sum_{i=1}^{d} |f(u)_i + f(u')_i|}$$

Here, $f(u)_i$ is the i-th element of vector $f(u)$.

For evaluation metrics, we first use the AUC, which measures the area under the ROC curve. The ROC curve projects the relation between false-positive rate (on the x-axis) and true-positive rate (on the y-axis) over a series of thresholds for a given prediction task. A ROC curve closer to the top-left border of the plot (high true-positive rate for low false-positive rate), thus a larger AUC value, indicates higher prediction performance. Morever, there exists a conventional standard to interpret AUC values:[5] AUC = 0.5 is equivalent to random guessing, whereas an AUC greater than 0.9 implies an excellent prediction. Many recent works on assessing privacy risks have adopted AUC as the evaluation metric [4], [37], [15], [13], [41], [21]. We also make use of the F1 score for the method that automatically detects fake edges. Due to the randomness of the anonymization alogrithms, we repeat our experiments five times and report the average results.

Parameters in Anonymization Mechanisms: We rely on SecGraph to perform k-DA and SalaDP [18]. Each anonymization mechanism has its own privacy parameter. For k-DA, we need to choose the value k, i.e., the minimal number of users sharing a certain degree for all possible degrees in $\mathcal{G}_\mathcal{A}$. Greater k implies stronger privacy. In our experiments, we choose k to be 50, 75, and 100, respectively, to explore different levels of privacy protection [18]. For SalaDP, the privacy parameter is ϵ which controls the noise added to the dK-2 series of $\mathcal{G}$: The smaller ϵ is, the higher its privacy provision is. Following previous works [40], [18], we experiment with three different ϵ values: 10, 50, and 100. As stated before, both k-DA and SalaDP's principal operation is adding fake edges to the original graph. By running the two anonymization mechanisms on our three datasets, we discover that this is indeed the case. For instance, SalaDP ($\epsilon = 10$) adds 120% more edges to the NO dataset, while only deleting 1.7% of the original edges.

Hyperparameter Setting: There are mainly three hyperparameters in the graph embedding phase: walk length (l), walk times (t) and vector dimension (d). For both k-DA and SalaDP, we choose $l = 100$ and $t = 80$. Meanwhile, we set $d = 128$ for k-DA and $d = 512$ for SalaDP. These values are selected through cross validation (see Section IV-C). For reproducibility purposes, our source code will be made publicly available.

B. Prediction Results

Figure 3 depicts the AUC values of using our edge plausibility metric (Cosine in Figure 3) to differentiate fake edges from original ones. In most of the cases, we achieve excellent performance with AUC values above 0.95. In particular, for the SalaDP-anonymized SNAP dataset ($\epsilon = 100$), the average AUC value is 0.971 (see Figure 3d). The only case where our edge plausibility does not achieve an excellent performance is when applying SalaDP on the Enron dataset where the AUC values are between 0.76 and 0.83. However, we emphasize that for most of the classification tasks, such AUC is already considered good.

We also notice that our method performs better against SalaDP on the SNAP dataset than the the other two. One reason is that SNAP has the highest number of average degrees (Table II), which implies more diverse dK-2 series. This

results in SalaDP adding more fake edges on SNAP, which leads to high performance of fake edge detection. However, we do not observe a similar trend for k-DA-anonymized datasets.

The AUC values for other vector similarity (distance) metrics are presented in Figure 3 as well. Cosine similarity performs slightly better than both Euclidean distance and Bray-Curtis distance on k-DA-anonymized graphs. On the other hand, for SalaDP-anonymized graphs, we can observe that cosine similarity performs better than Euclidean distance (around 10% performance gain), while the performance of Bray-Curtis and cosine similarity is still very close. This shows that cosine similarity (as well as Bray-Curtis distance) is a suitable choice for our edge plausibility metric.

Figure 3 also shows that our edge plausibility significantly outperforms the traditional structural proximity metrics. For instance, on the SalaDP-anonymized NO dataset ($\epsilon = 50$), our approach achieves 0.944 AUC while the result for the best performing structural proximity, i.e., Jaccard index, is around 0.7. It also appears that embeddedness outperforms the other two metrics on k-DA-anonymized dataset in most of the cases, while Jaccard index is rather effective for SalaDP.

C. Hyperparameter Sensitivity

We study the influences of the three hyperparameters (l, t and d) on the prediction performance. Here, l and t are directly related to the size of the random walk traces, which essentially decides the amount of data used for learning embedding vectors. For both anonymization mechanisms, we observe that increasing l and t improves the AUC values. However, the increase is smaller when both of these values are above 60. Therefore, we set $l = 100$ and $t = 80$.

Meanwhile, we observe interesting results for the vector dimension d: different anonymization mechanisms have different optimal choices for d (Figure 4 and Figure 5). It appears that when detecting fake edges on k-DA-anonymized graphs, $d = 128$ is a suitable choice for all datasets. On the other hand, for SalaDP, $d = 512$ is able to achieve a stronger prediction. We confirm that the vector dimension is indeed a subtle parameter, as was observed in other data domains, such as biomedical data [3] and mobility data [4]. In conclusion, our default hyperparameter settings are suitable for our prediction task.

D. Optimizing Fake Edge Detection

Next, we investigate how to concretely determine whether an edge in an anonymized graph is fake given its plausibility, such that the adversary can recover the original graph from the anonymized one.

Figure 1a and Figure 6 depict the histograms of both fake and original edges' plausibility in anonymized NO dataset (by both k-DA and SalaDP). We see that both follow a Gaussian distribution with different means and standard deviations. Similar results are observed on Enron and SNAP datasets.

Given that the general population (plausibility of all edges) consists of a mixture of two subpopulations (plausibility of fake and original edges) with each one following a Gaussian distribution, we can fit the general population with a Gaussian mixture model (GMM). With the fitted GMM, we can obtain
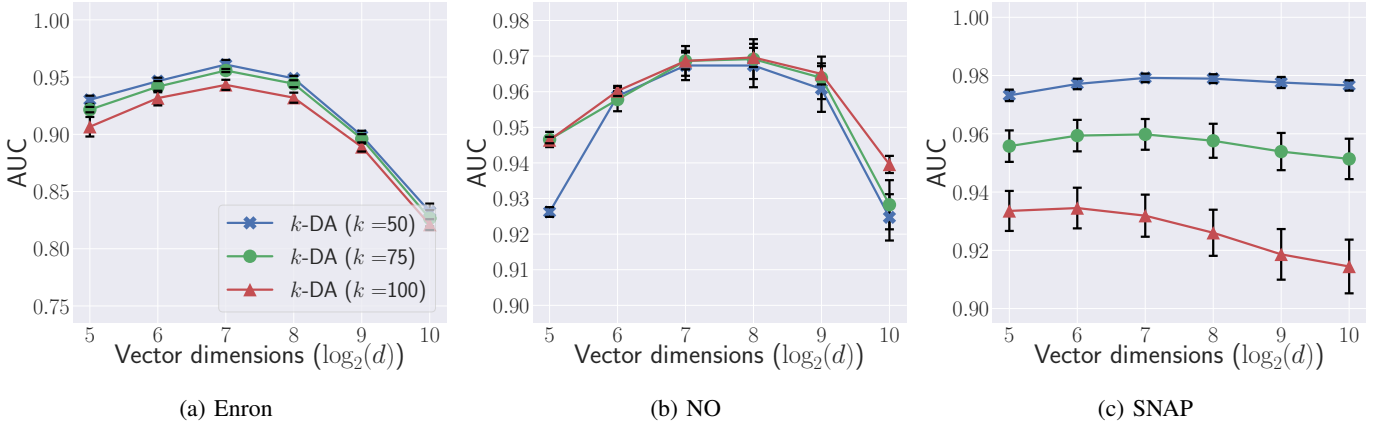

Fig. 4: [Higher is better] Sensitivity of the AUC with respect to the embedding vector dimension for k-DA-anonymized datasets.
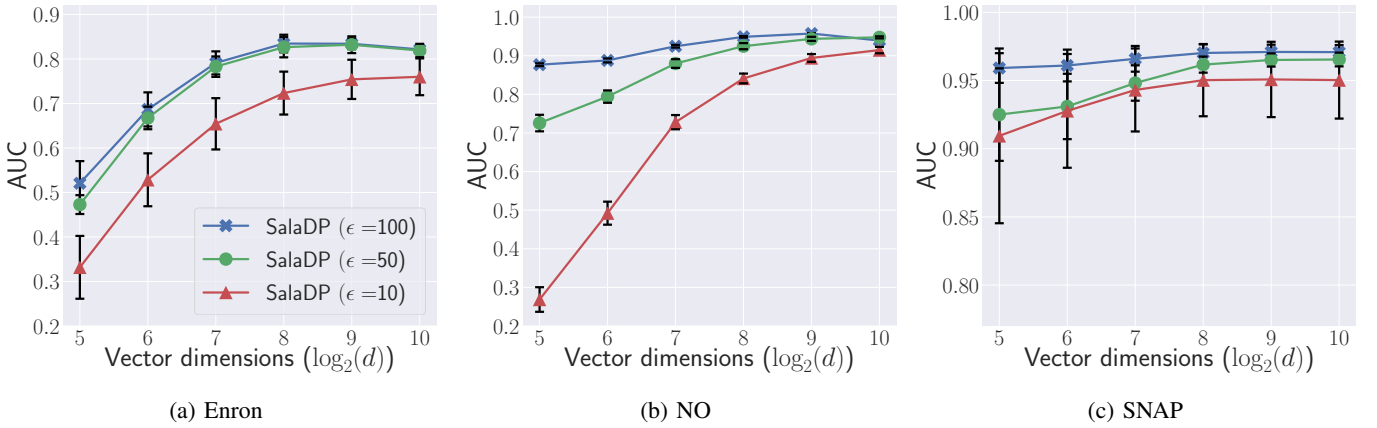

Fig. 5: [Higher is better] Sensitivity of the AUC with respect to the embedding vector dimension for SalaDP-anonymized datasets.

each edge's posterior of being fake or original given its plausibility. If the former is higher than the latter, then we predict the edge to be fake, effectively performing a maximum a posteriori (MAP) estimate. This means GMM and MAP estimate provide us with a concrete threshold to determine whether an edge is fake given the observed data.

Gaussian Mixture Model: To formally define our GMM, we first introduce two random variables: $\mathcal{B}$ and $\mathcal{S}$. $\mathcal{B}$ represents whether an edge is original ($\mathcal{B} = 0$) or fake ($\mathcal{B} = 1$), while $\mathcal{S}$ represents the plausibility of an edge. The probability density function of our GMM is formally defined as:

$$p(\mathcal{S} = s_{\mathcal{A}}(u, u')) = \sum_{i \in \{0,1\}} w_i N(s_{\mathcal{A}}(u, u')|\mu_i, \sigma_i).$$

The GMM is parametrized by 6 parameters: w_0, μ_0, σ_0, w_1, μ_1 and σ_1. Here, w_0 (w_1) is the prior probability of an edge being original (fake), i.e., $w_0 = P(\mathcal{B} = 0)$ ($w_1 = P(\mathcal{B} = 1)$). The other 4 parameters are related to the two Gaussian distributions for edge plausibility: $N(s_{\mathcal{A}}(u, u')|\mu_i, \sigma_i)$ for $i \in \{0,1\}$ is the density function of the Gaussian distribution:

$$\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{1}{2\sigma_i^2}(s_{\mathcal{A}}(u, u') - \mu_i)).$$

Parameter Learning: To learn the 6 parameters of the GMM, we adopt the expectation maximization (EM) algorithm, which consists of two steps, i.e., the expectation (E) step and the maximization (M) step. The E-step calculates, for each edge in $\mathcal{G}_{\mathcal{A}}$, its posterior probability of being fake or original given its plausibility value. The M-step updates all the 6 parameters based on the probabilities calculated from the E-step following maximum likelihood estimation. The learning process iterates over the two steps until convergence. Here, convergence means that the log-likelihoods of two consecutive iterations differ less than a given threshold (we set it to 0.001 in our experiments). In addition, the initial values of the 6 parameters are set randomly.

Fake Edge Detection: After the GMM has been learned, we compute for each edge $\{u, u'\}$ its posterior probabilities of being original and fake:

$$P(\mathcal{B} = 0|s_{\mathcal{A}}(u, u')) = \frac{w_0 N(s_{\mathcal{A}}(u, u')|\mu_0, \sigma_0)}{\sum_{i \in \{0,1\}} w_i N(s_{\mathcal{A}}(u, u')|\mu_i, \sigma_i)}$$

$$P(\mathcal{B} = 1|s_{\mathcal{A}}(u, u')) = \frac{w_1 N(s_{\mathcal{A}}(u, u')|\mu_1, \sigma_1)}{\sum_{i \in \{0,1\}} w_i N(s_{\mathcal{A}}(u, u')|\mu_i, \sigma_i)}$$

(a) k-DA ($k = 50$)

(b) k-DA ($k = 75$)

(c) SalaDP ($\epsilon = 100$)

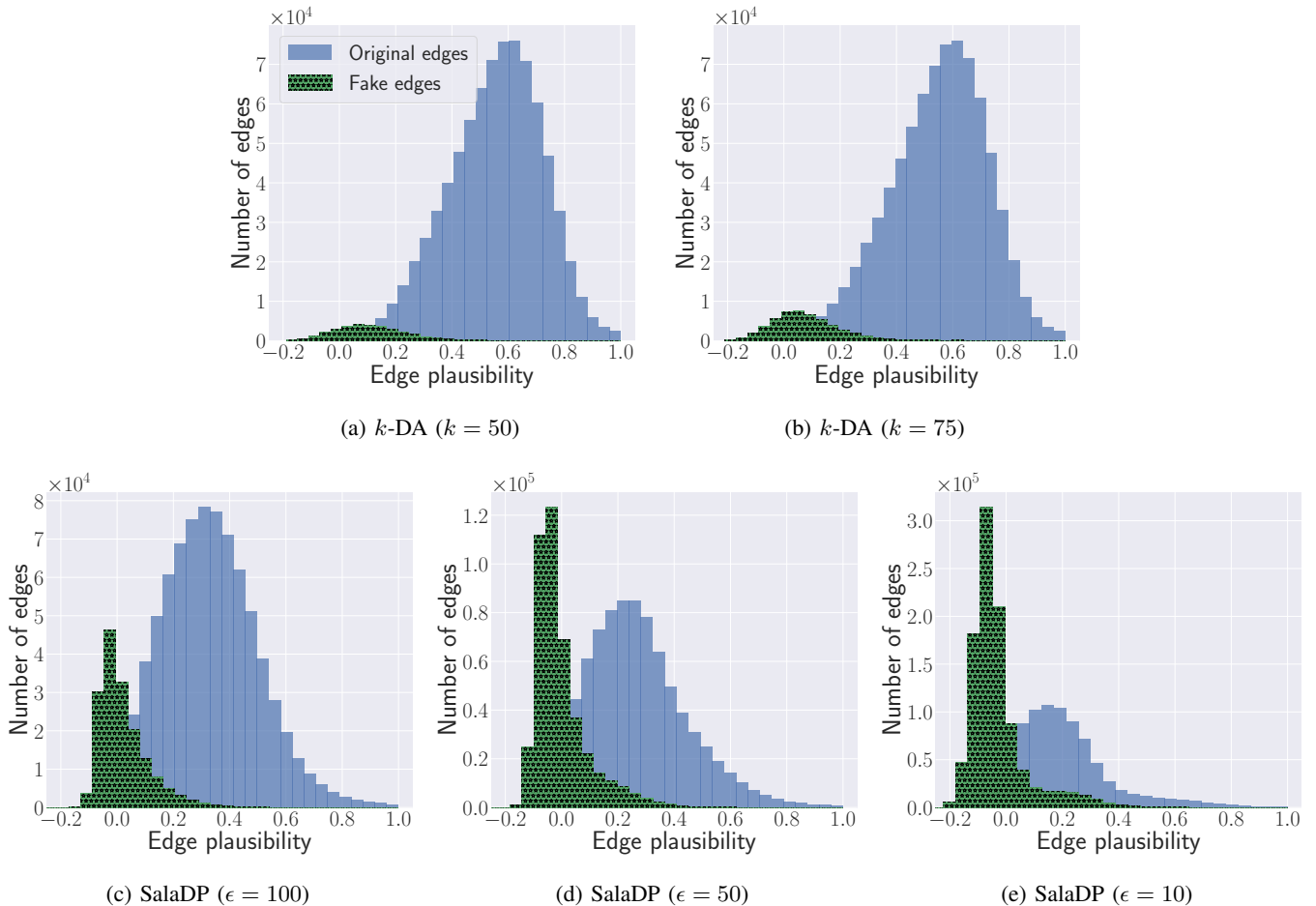(d) SalaDP ($\epsilon = 50$)

(e) SalaDP ($\epsilon = 10$)

Fig. 6: Plausibility distributions of fake and original edges in the NO dataset anonymized by the two anonymization mechanisms. The result for k-DA ($k = 100$) is depicted in Figure 1a.

TABLE III: [Higher is better] F1 scores for detecting fake edges using GMM and MAP estimate for both k-DA and SalaDP on three different datasets.

	Enron	NO	SNAP
k-DA ($k = 50$)	0.792	0.642	0.857
k-DA ($k = 75$)	0.796	0.710	0.869
k-DA ($k = 100$)	0.812	0.761	0.881
SalaDP ($\epsilon = 100$)	0.672	0.712	0.853
SalaDP ($\epsilon = 50$)	0.750	0.723	0.835
SalaDP ($\epsilon = 10$)	0.819	0.876	0.802

and pick the one that is maximum (MAP estimate): If $P(\mathcal{B} = 1|s_{\mathcal{A}}(u, u')) > P(\mathcal{B} = 0|s_{\mathcal{A}}(u, u'))$, we predict $\{u, u'\}$ to be fake, and vice versa.

In the end, we delete all the predicted fake edges, and obtain the recovered graph $\mathcal{G}_{\mathcal{R}}$.

Results: We train GMMs under both anonymization mechanisms for all the datasets. Table III presents the results. We first observe that, in most of the cases, our approach achieves a strong prediction, e.g., for the SalaDP-anonymized NO dataset

($\epsilon = 10$), the F1 score is 0.876. For our worst prediction on SalaDP-anonymized Enron dataset ($\epsilon = 100$), the F1 score is still approaching 0.7. Another interesting observation is that when the privacy level increases, i.e., higher k or lower ϵ, our prediction performance increases in most of the cases. This can be explained by the fact that higher privacy levels lead to more fake edges being added.

V. PRIVACY LOSS

As fake edges help an anonymized graph $\mathcal{G}_{\mathcal{A}}$ satisfy certain privacy guarantees, we expect that, by obtaining the recovered graph $\mathcal{G}_{\mathcal{R}}$ from $\mathcal{G}_{\mathcal{A}}$, these guarantees will be violated. In this section, we first define two metrics tailored to each anonymization mechanism for quantifying the privacy loss due to our graph recovery attack. Then, we present the corresponding evaluation results.

A. Privacy Loss Measurement

k-**DA:** k-DA assumes that the adversary only has knowledge of her targets' degrees and uses this knowledge to re-identify them. This means that, if the users' degrees in $\mathcal{G}_{\mathcal{R}}$ are more similar to those in $\mathcal{G}$ compared to $\mathcal{G}_{\mathcal{A}}$, then the adversary is

more likely to achieve her goal. Therefore, we propose to compute users' average degree difference between $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}$, as well as between $\mathcal{G}_\mathcal{R}$ and $\mathcal{G}$, to measure the privacy loss caused by our graph recovery. Formally, we define users' average degree difference between $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}$ as:

$$\Delta_\mathcal{A} = \frac{\sum_{u \in \mathcal{U}} ||\kappa(u)| - |\kappa_\mathcal{A}(u)||}{|\mathcal{U}|}$$

and define users' average degree difference between $\mathcal{G}$ and $\mathcal{G}_\mathcal{R}$ ($\Delta_\mathcal{R}$) accordingly.

Note that our approach also deletes some original edges when recovering $\mathcal{G}_\mathcal{R}$ (false positives). Therefore, if the adversary relies on the users' exact degrees (as assumed in k-DA) to de-anonymize them, she might fail. However, a sophisticated adversary can apply extra heuristics, such as tolerating some degree differences for finding her targets. In this case, $\Delta_\mathcal{R}$ being smaller than $\Delta_\mathcal{A}$ can still provide the adversary with a better chance to achieve her goal.

SalaDP: To quantify the privacy loss for SalaDP, we consider the noise added to the dK-2 series of the original graph $\mathcal{G}$. Formally, the dK-2 series of $\mathcal{G}$, denoted by $D(\mathcal{G})$, is a set with each element $r_{i,j}(\mathcal{G})$ in $D(\mathcal{G})$ representing the number of edges that connect users of degrees i and j in $\mathcal{G}$. Formally, $r_{i,j}(\mathcal{G})$ is defined as:

$$r_{i,j}(\mathcal{G}) = |\{\{u, u'\}|\{u, u'\} \in \mathcal{E} \wedge |\kappa(u)| = i \wedge |\kappa(u')| = j\}|.$$

Accordingly, $r_{i,j}(\mathcal{G}_\mathcal{A})$ and $r_{i,j}(\mathcal{G}_\mathcal{R})$ represent the corresponding numbers in $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}_\mathcal{R}$. Then, we use $\zeta_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{A}) = r_{i,j}(\mathcal{G}_\mathcal{A}) - r_{i,j}(\mathcal{G})$ to denote the noise added to $r_{i,j}(\mathcal{G})$ when transforming $\mathcal{G}$ to $\mathcal{G}_\mathcal{A}$, and $\zeta_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{R}) = r_{i,j}(\mathcal{G}_\mathcal{R}) - r_{i,j}(\mathcal{G})$ to represent the (lower) noise caused by our graph recovery attack. Since SalaDP is a statistical mechanism, we sample 100 anonymized graphs $\{\mathcal{G}_\mathcal{A}^t\}_{t=1}^{100}$ by applying SalaDP to $\mathcal{G}$ 100 times and produce 100 noise samples $\{\zeta_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{A}^t)\}_{t=1}^{100}$ for each element in $D(\mathcal{G})$.

We define two metrics for quantifying the privacy loss due to graph recovery for SalaDP. In the first metric, we compare the difference of average noise added to $D(\mathcal{G})$ before (by SalaDP) and after graph recovery. Concretely, for each $r_{i,j}(\mathcal{G})$ in $D(\mathcal{G})$, we first calculate the average absolute noise added to $r_{i,j}(\mathcal{G})$, denoted by $\bar{\zeta}_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{A})$, over the 100 SalaDP graph samples described above, i.e.,

$$\bar{\zeta}_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{A}) = \frac{\sum_{t=1}^{100} |\zeta_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{A}^t)|}{100}.$$

Then, we compute the average noise over the whole graph as:

$$\zeta_\mathcal{A} = \frac{1}{|D(\mathcal{G})|} \sum_{r_{i,j}(\mathcal{G}) \in D(\mathcal{G})} \bar{\zeta}_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{A}).$$

We analogously compute the average added noise $\zeta_\mathcal{R}$ after our graph recovery attack.

For the second approach, we consider the uncertainty introduced by the added noise. McGregor et al. explore the connection between privacy and the uncertainty of the output produced by differential privacy mechanisms [28]. In general, higher uncertainty implies stronger privacy. We measure the uncertainty of noise added by SalaDP through estimating its

TABLE IV: Differences in average degree between the original graph ($\mathcal{G}$), the k-DA anonymized graph ($\mathcal{G}_\mathcal{A}$) and our recovered graph ($\mathcal{G}_\mathcal{R}$).

	Enron		NO		SNAP	
	$\Delta_\mathcal{R}$	$\Delta_\mathcal{A}$	$\Delta_\mathcal{R}$	$\Delta_\mathcal{A}$	$\Delta_\mathcal{R}$	$\Delta_\mathcal{A}$
k-DA ($k = 50$)	0.990	1.222	0.499	0.541	6.589	8.216
k-DA ($k = 75$)	1.367	1.705	0.752	0.875	8.815	11.755
k-DA ($k = 100$)	2.019	2.377	1.035	1.231	11.565	16.018

empirical entropy. To this end, we calculate the Shannon entropy over the frequencies of elements in $\{\zeta_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{A}^t)\}_{t=1}^{100}$ (the 100 noise samples described above), denoted by $\hat{H}_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{A})$. Then, we compute the average entropy of the noise as follows:

$$\hat{H}_\mathcal{A} = \frac{1}{|D(\mathcal{G})|} \sum_{r_{i,j}(\mathcal{G}) \in D(\mathcal{G})} \hat{H}_{i,j}(\mathcal{G}, \mathcal{G}_\mathcal{A}).$$

We compute the average entropy after our graph recovery similarly and denote it as $\hat{H}_\mathcal{R}$.

B. Evaluation

k-DA: Table IV presents the results of the users' degree differences. In all cases, $\Delta_\mathcal{R}$ is smaller than $\Delta_\mathcal{A}$. This indicates that the adversary has a better chance to identify her targets from $\mathcal{G}_\mathcal{R}$ than from $\mathcal{G}_\mathcal{A}$, and demonstrates that our attack clearly decreases the privacy provided by k-DA. It also appears that our graph recovery gains least benefits for the adversary on the NO dataset, where $\Delta_\mathcal{R}$ is closer to $\Delta_\mathcal{A}$. This is essentially due to the fact that the original NO dataset already preserves a high k-degree anonymity.

SalaDP: Table V presents the average noise added to the dK-2 series of the original graph with respect to the anonymized and recovered graphs. We observe that, in all cases, $\zeta_\mathcal{R}$ is smaller than $\zeta_\mathcal{A}$ showing that our recovery attack reduces the average noise for SalaDP. We also observe that the relative reduction of the average noise with our graph recovery in general decreases when increasing ϵ: The added noise is already much smaller for larger ϵ and cannot be further reduced.

Table VI presents the average entropy of the noise added to the dK-2 series of the original graph after applying SalaDP and after the graph recovery attack. Note that, while one would expect higher entropy for smaller values of ϵ, this does not hold true in practice because the SalaDP mechanism is not necessarily optimal with respect to the added uncertainty. Still, across all values of ϵ and all the datasets, we can observe a reduction of the empirical entropy, and therefore a reduction of the privacy provision. The relative reduction, however, varies between the values of ϵ and, as for the average noise above, between the datasets.

For now, it seems unclear how these various factors impact the relative reduction of empirical entropy. Analyzing the impact of these parameters on the relative reduction of empirical entropy could provide further insights into the recoverability of anonymized graphs. Such work is, however, orthogonal to the work presented in this paper and could be an interesting direction for our future work.

TABLE V: Differences in average noise on the original graph $\mathcal{G}$'s dK-2 series by SalaDP ($\zeta_\mathcal{A}$) and by our graph recovery attack ($\zeta_\mathcal{R}$).

	Enron		NO		SNAP	
	$\zeta_\mathcal{R}$	$\zeta_\mathcal{A}$	$\zeta_\mathcal{R}$	$\zeta_\mathcal{A}$	$\zeta_\mathcal{R}$	$\zeta_\mathcal{A}$
SalaDP ($\epsilon = 100$)	4.432	5.282	6.048	6.415	3.422	4.018
SalaDP ($\epsilon = 50$)	4.224	7.121	7.731	9.471	3.489	4.445
SalaDP ($\epsilon = 10$)	4.958	12.004	7.982	16.033	3.672	5.690

TABLE VI: Differences in average entropy of the noise on the original graph $\mathcal{G}$'s dK-2 series by SalaDP ($\hat{H}_\mathcal{A}$) and by our graph recovery attack ($\hat{H}_\mathcal{R}$).

	Enron		NO		SNAP	
	$\hat{H}_\mathcal{R}$	$\hat{H}_\mathcal{A}$	$\hat{H}_\mathcal{R}$	$\hat{H}_\mathcal{A}$	$\hat{H}_\mathcal{R}$	$\hat{H}_\mathcal{A}$
SalaDP ($\epsilon = 100$)	0.180	2.029	1.243	2.515	1.999	2.209
SalaDP ($\epsilon = 50$)	0.556	1.865	1.754	2.852	2.000	2.238
SalaDP ($\epsilon = 10$)	1.095	1.381	2.275	3.112	1.926	2.022

Note that differential privacy guarantees are theoretically not violated since differential privacy is, by definition, closed under post-processing [11]. However, despite these formal semantic guarantees are still valid, we demonstrate that our recovery attack can, without additional knowledge about the data, reduce the magnitude of the actual noise put in place to perturb the original graph by exploiting the graph structure. This demonstrates that, by simply looking at the sanitized data, we can concretely jeopardize the anonymity of the graph.

Graph De-anonymization: We also compare the performance of the graph de-anonymization attack designed by Narayanan and Shmatikov [32], referred to as the NS-attack, on both anonymized and recovered social graphs. Our experiments show that, contrary to what one might initially expect, graph recovery does not improve the performance of the graph de-anonymization significantly. Our explanation is that the NS-attack assumes a much stronger adversary model, such as an auxiliary graph with seed nodes already de-anonymized (see Section VI). Moreover, Ji et al. show that, in many cases, the NS-attack even performs better on the anonymized graph than on the original graph [18].

VI. ENHANCING GRAPH ANONYMIZATION

In this section, we take the first step towards enhancing the existing graph anonymization mechanisms. We start by presenting our methodology, then evaluate the performance of fake edge detection as well as graph utility with the enhanced mechanisms. In the end, we study our new anonymized graphs' resistance to graph de-anonymization.

A. Methodology

To improve the graph anonymization mechanisms, intuitively, we should add fake edges that are more similar to edges in the original graph $\mathcal{G}$. Figure 7 depicts the edge plausibility distributions for the original NO dataset under two different vector dimensions.[6] We observe that both empirical
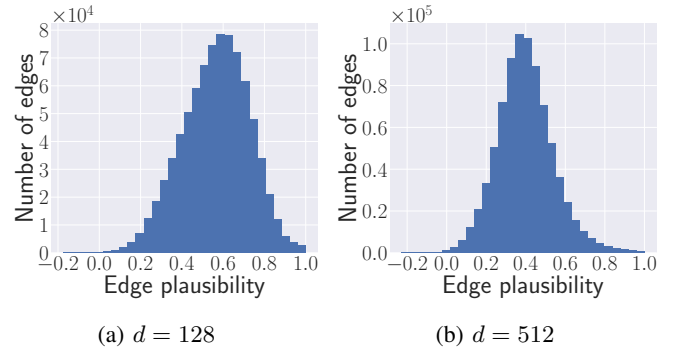

(a) $d = 128$ (b) $d = 512$

Fig. 7: Edge plausibility in the original NO dataset follows a Gaussian distribution. We choose two vector dimensions for edge plausibility: (a) $d = 128$ and (b) $d = 512$, following the evaluation results in Section IV.

distributions follow a Gaussian distribution. If we are able to modify the current graph anonymization mechanisms such that the plausibility of the added fake edges is more likely to come from the same Gaussian distribution, it should be harder to discover these fake edges.

The general procedure for our enhanced anonymization mechanisms is as follows. We first apply maximum likelihood estimation to learn the Gaussian distribution of edge plausibility in $\mathcal{G}$, denoted by $N(s(u, u')|\mu, \sigma)$, where $s(u, u')$ represents $\{u, u'\}$'s plausibility in $\mathcal{G}$. Then, we conduct the same process as in k-DA and SalaDP. A loop is performed through all the users and, in each iteration, if a user u needs m fake edges, we construct a candidate set $\gamma(u)$ which includes all the potential users that could share a fake edge with u (following the original anonymization mechanisms' design). Different from the original approaches of k-DA and SalaDP for choosing m users out of $\gamma(u)$, we compute the plausibility between users in $\gamma(u)$ and u,[7] represented as a set $\lambda(u) = \{s(u, v)|v \in \gamma(u)\}$. Then, for each plausibility $s(u, v)$ in $\lambda(u)$, we calculate its density using the previously learned $N(s(u, u')|\mu, \sigma)$, and treat the density as the *weight* of the user v in $\gamma(u)$. Next, we perform a weighted sampling to choose m users out of $\gamma(u)$ and add edges between these users and u. In the end, we obtain our new anonymized graph $\mathcal{G}_\mathcal{F}$ under the enhanced mechanisms.

Note that, as presented in Section II, for a user u, SalaDP chooses m users from $\gamma(u)$ in a random manner, while k-DA picks the users with the highest residual degrees. However, the reason for k-DA to take this approach is to efficiently construct the anonymized graph. Through experiments, we discover that our enhanced k-DA can also build the anonymized graph in a similar time.

We emphasize that our enhanced mechanisms do not affect the privacy criteria of k-DA and SalaDP as they do not modify the privacy realization process of the original mechanisms. We will make the source code for the aforementioned enhanced versions of k-DA and SalaDP publicly available.

---

[6]We map all users in $\mathcal{G}$ into vectors and compute all edges' plausibility in $\mathcal{G}$ following the same procedure as for $\mathcal{G}_\mathcal{A}$ (Section III).

[7]The plausibility is computed over users' vectors learned from $\mathcal{G}$.

(a) Enhanced k-DA ($k = 50$)

(b) Enhanced k-DA ($k = 75$)

(c) Enhanced SalaDP ($\epsilon = 100$)

(d) Enhanced SalaDP ($\epsilon = 50$)
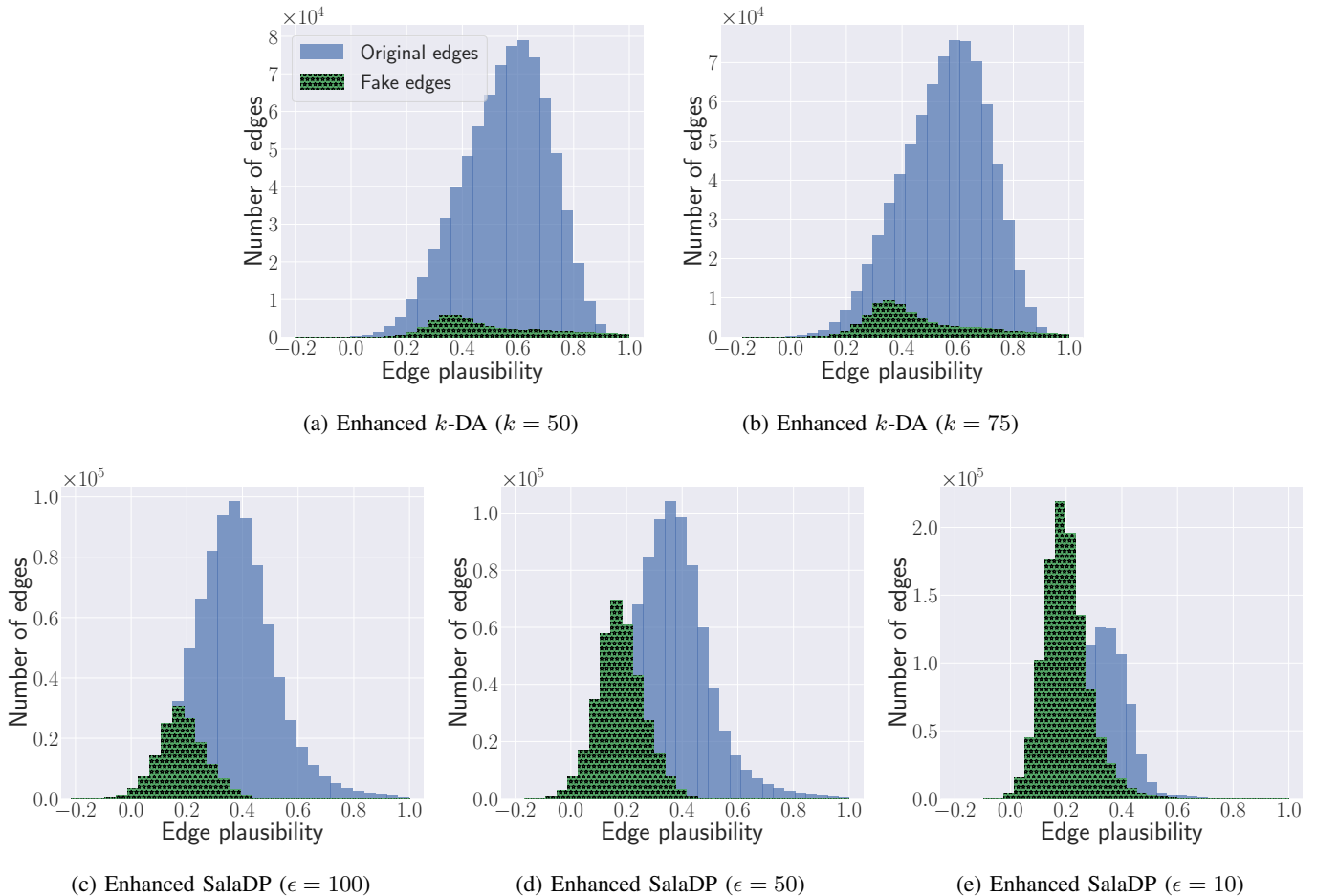

(e) Enhanced SalaDP ($\epsilon = 10$)

Fig. 8: Plausibility distributions of fake and original edges in the NO dataset anonymized with our enhanced mechanisms. The result for k-DA ($k = 100$) is depicted in Figure 1b.

TABLE VII: [Higher is better] AUC scores for detecting fake edges for both enhanced k-DA and SalaDP on three different datasets.

	Enron	NO	SNAP
k-DA ($k = 50$)	0.677	0.628	0.939
k-DA ($k = 75$)	0.728	0.676	0.927
k-DA ($k = 100$)	0.753	0.702	0.896
SalaDP ($\epsilon = 100$)	0.806	0.890	0.719
SalaDP ($\epsilon = 50$)	0.794	0.895	0.723
SalaDP ($\epsilon = 10$)	0.724	0.853	0.723

TABLE VIII: [Higher is better] F1 scores for detecting fake edges using GMM and MAP estimate for both enhanced k-DA and SalaDP on three different datasets.

	Enron	NO	SNAP
k-DA ($k = 50$)	0.531	0.391	0.632
k-DA ($k = 75$)	0.428	0.433	0.609
k-DA ($k = 100$)	0.510	0.501	0.597
SalaDP ($\epsilon = 100$)	0.422	0.370	0.515
SalaDP ($\epsilon = 50$)	0.390	0.411	0.522
SalaDP ($\epsilon = 10$)	0.439	0.527	0.490

B. Evaluation

Fake Edge Detection: After obtaining $\mathcal{G}_{\mathcal{F}}$, we perform the same process as in Section III to compute the plausibility of all edges in $\mathcal{G}_{\mathcal{F}}$. Then, we calculate the AUC values when using plausibility to differentiate between fake and original edges in $\mathcal{G}_{\mathcal{F}}$. The results are presented in Table VII.

First of all, the AUC values drop in all cases compared to the results in Figure 3. Especially for the k-DA-anonymized NO dataset ($k = 50$), AUC drops by 35% to 0.628. This can be

also observed from the histograms in Figure 1b and Figure 8: By plausibility, fake edges are hidden quite well among the original edges (compared to Figure 1a and Figure 6). When applying our enhanced k-DA mechanism on SNAP, the AUC values drop, but less than for NO. This may be due to the dataset's small size (4,039 users) and the large k value, which leads to a large number of fake edges. On the other hand, the performance decrease for SalaDP-anonymized datasets is smaller, but still significant.

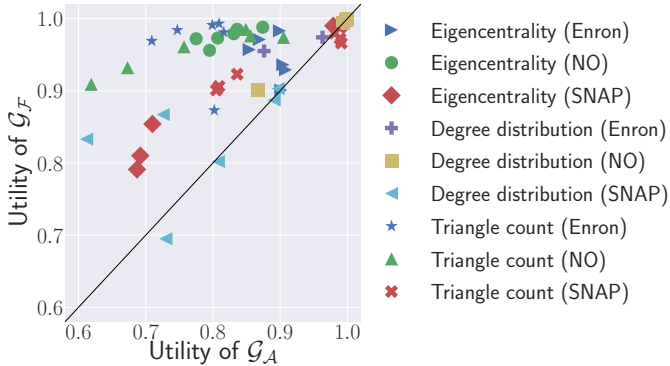Moreover, we discover from Figure 1b and Figure 8 that the

Fig. 9: Comparing the utility of our enhanced mechanism ($\mathcal{G}_{\mathcal{F}}$) to the original mechanism ($\mathcal{G}_{\mathcal{A}}$) for different datasets and metrics. Any point above the diagonal indicates better utility of our anonymized graph. The x-axis is the cosine similarity of $\mathcal{G}_{\mathcal{A}}$ to the original graph $\mathcal{G}$, and the y-axis analogue for $\mathcal{G}_{\mathcal{F}}$.

two Gaussian distributions of $\mathcal{G}_{\mathcal{F}}$ for k-DA and SalaDP largely overlap (see Figure 1a and Figure 6 for comparison). This indicates that the Gaussian mixture model approach described in Section IV cannot perform effective fake edge detection. For instance, our experiments with the GMM approach only achieve around 0.37 F1 score for SalaDP ($\epsilon = 100$) on the NO dataset, which represents a 50% performance drop (see Table VIII).

It is worth noting that all the edges added by our enhanced anonymization mechanisms still have relatively smaller plausibility than the original edges. Given that our weighted sampling follows the original edges' plausibility distribution in $\mathcal{G}$, this implies that not many potential fake edges are normal with respect to plausibility. We conclude that it is non-trivial to create fake edges totally indistinguishable from original edges.

Graph Utility: The main motivation for OSNs to share their graph data is to allow third parties to conduct research or build commercial applications. Therefore, a graph anonymization mechanism needs to take into account graph utility, i.e., how well the anonymized graph preserves the structural properties of the original graph. To show that our enhanced mechanisms outperform the current anonymization mechanisms, we also evaluate $\mathcal{G}_{\mathcal{F}}$'s utility.

There exist many graph properties that can be used to evaluate graph utility [12], [24], [18]. For the sake of conciseness, we focus on three of them including degree distribution, eigencentrality, and triangle count. The degree distribution represents the proportion of users with a certain degree for all possible degrees. Eigencentrality evaluates the influence/importance of each user in a graph. It assigns a centrality score for each user based on the eigenvector of the graph's adjacency matrix. Triangle count summarizes the number of triangles each user belongs to in a graph which reflects the graph connectivity [24]. We compute the three properties for $\mathcal{G}$, $\mathcal{G}_{\mathcal{A}}$, and $\mathcal{G}_{\mathcal{F}}$, and calculate the cosine similarity between $\mathcal{G}$'s and $\mathcal{G}_{\mathcal{A}}$'s properties as well as between $\mathcal{G}$'s and $\mathcal{G}_{\mathcal{F}}$'s properties. Higher similarity naturally implies better utility.

Figure 9 presents the results. We first observe a strong

TABLE IX: De-anonymization prevention of our enhanced mechanism ($\mathcal{G}_{\mathcal{F}}$) and the original mechanism ($\mathcal{G}_{\mathcal{A}}$). [Lower is better] Number of nodes the NS-attack can correctly de-anonymize. Best scores are in bold.

	Enron		NO		SNAP	
	$\mathcal{G}_{\mathcal{A}}$	$\mathcal{G}_{\mathcal{F}}$	$\mathcal{G}_{\mathcal{A}}$	$\mathcal{G}_{\mathcal{F}}$	$\mathcal{G}_{\mathcal{A}}$	$\mathcal{G}_{\mathcal{F}}$
k-DA ($k = 50$)	307	289	759	**532**	328	303
k-DA ($k = 75$)	309	270	689	508	294	234
k-DA ($k = 100$)	302	**256**	580	491	274	**208**
SalaDP ($\epsilon = 100$)	265	255	470	**396**	378	342
SalaDP ($\epsilon = 50$)	243	225	291	277	370	290
SalaDP ($\epsilon = 10$)	236	**207**	233	208	376	**267**

similarity between $\mathcal{G}_{\mathcal{F}}$ and $\mathcal{G}$ for all graph properties, i.e., $\mathcal{G}_{\mathcal{F}}$ preserves high utility. For instance, the cosine similarity for triangle count is above 0.86 in most of the cases. Meanwhile, the lowest cosine similarity (degree distribution) is still approaching 0.7 when applying enhanced k-DA ($k = 100$) to SNAP.

More importantly, we observe that $\mathcal{G}_{\mathcal{F}}$ preserves better graph utility than $\mathcal{G}_{\mathcal{A}}$ (almost all points in Figure 9 are above the diagonal). For instance, the eigencentrality's cosine similarity between $\mathcal{G}_{\mathcal{F}}$ and $\mathcal{G}$ is 0.985 while the similarity between $\mathcal{G}_{\mathcal{A}}$ and $\mathcal{G}$ is only 0.836 for the k-DA-anonymized NO dataset ($k = 50$). This is because the fake edges added by our enhanced mechanisms are more structurally similar to the original edges, thus preserving better utility.

Graph De-anonymization: Next, we investigate the performance of graph de-anonymization on graphs generated by our enhanced mechanisms. We concentrate on the NS-attack [32] due to its superior performance over others [18]. The NS-attack assumes that the adversary knows an *auxiliary graph* with all nodes' identities. Her goal is to map each node in the auxiliary graph to the node representing the same user in an anonymized *target graph*. Correctly matched nodes are thus successfully de-anonymized in the target graph. To ease this matching, the NS-attack assumes that the adversary has prior knowledge of some correctly matched nodes, namely the *seed nodes*. The attack then starts from these seeds to de-anonymize more nodes by propagating throughout the whole anonymized graph.

We use $\mathcal{G}_{\mathcal{A}}$ and $\mathcal{G}_{\mathcal{F}}$ as the target graphs, respectively, and sample a subgraph from the original graph $\mathcal{G}$ containing all edges among 25% randomly selected nodes in $\mathcal{G}$ as the auxiliary graph. Moreover, we choose the 200 nodes with the highest degrees from the auxiliary graph as our seeds.[8] For evaluation, we concentrate on correctly and wrongly de-anonymized users.

Table IX shows the results. First of all, the number of correctly de-anonymized nodes by the NS-attack is reduced in all cases thanks to our enhanced mechanisms. Figure 10 further depicts the anonymity gain, i.e., the performance drop with respect to the correctly de-anonymized nodes. We see that the NS-attack de-anonymizes almost 30% fewer nodes on the enhanced k-DA-anonymized (k=50) NO dataset. We also

---

[8]We tried other sampling approaches for seed nodes, but did not observe significant performance differences.
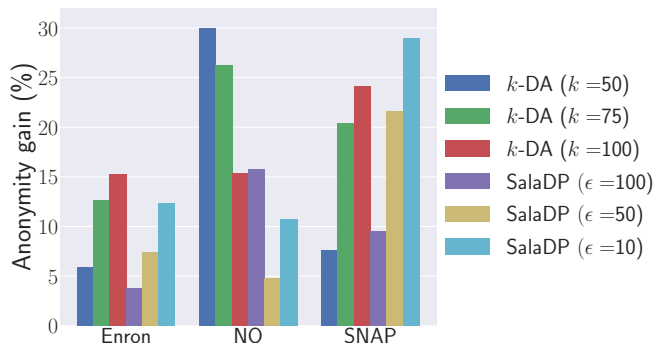
Fig. 10: [Higher is better] Gain in anonymity with respect to the reduction of the number of correctly de-anonymized nodes by the NS-attack between using our enhanced mechanism versus the original mechanism.

notice from Table IX that our enhanced mechanisms reduce the total number of nodes that the NS-attack de-anonymizes (both correct and wrong ones). This indicates that the NS-attack's ability to propagate also degrades in graphs anonymized by our enhanced mechanisms.

VII.   RELATED WORK

Various graph anonymization mechanisms have been proposed in the literature [26], [52], [17], [8], [53], [46], [9], [36], [48], [49], [31]. One class of these mechanisms follows the concept of k-anonymity. Liu and Terzi [26] propose the first approach in this direction, i.e., k-DA, which we concentrate on in this paper. Meanwhile, Zhou and Pei [52] propose k-neighborhood anonymity, where each user in the anonymized graph shares the same neighborhood, i.e., the sub-social network among her friends, with at least $k - 1$ other users. The authors adopt minimum BFS coding to represent each user's neighborhood, then rely on a greedy match to realize k-neighborhood anonymity.

Another class of graph anonymization mechanisms is inspired by differential privacy. Besides SalaDP, multiple solutions have been proposed [36], [48], [49]. For instance, Wang and Wu [48] present a 2K-graph generation model to achieve differential privacy, where noise is added based on smooth sensitivity. Xiao et al. [49] encode users' connection probabilities with a hierarchical random graph model, and perform Markov chain Monte Carlo to sample a possible graph structure from the model while enforcing differential privacy. Besides the above, other graph anonymization techniques include [17], [8], [31].

Due to space constraints, we only consider the two most widely known anonymization mechanisms, i.e., k-DA and SalaDP. In the future, we plan to apply our approach to more anonymization mechanisms.

Besides anonymization, graph de-anonymization has been extensively studied as well. Backstrom et al. are among the first to de-anonymize users in a naively anonymized social graph [5]. The attack of Narayanan and Shmatikov is essentially a framework [32], based on which multiple approaches

have been proposed [43], [19], [33], [42]. We emphasize that graph de-anonymization is orthogonal to our graph recovery attack. First of all, graph de-anonymization attacks aim to identify users in an anonymized graph while our graph recovery aims to find fake added edges. As shown in Section V, our graph recovery can degrade anonymized graphs' privacy guarantees. The reason our graph recovery cannot increase the performance of graph de-anonymization (in our case, of the NS-attack), is that most of the graph de-anonymization attacks assume a much stronger attack model than those considered in graph anonymization mechanisms. Therefore, we propose privacy loss metrics tailored to k-DA and SalaDP, which we believe are more appropriate. Moreover, we show that our enhanced anonymization mechanisms that are inspired by our graph recovery attack significantly reduce the success rate of graph de-anonymization.

VIII.   CONCLUSION

In this paper, we identify a fundamental vulnerability of the existing graph anonymization mechanisms which do not take into account key structural characteristics of a social graph when adding fake edges to it. We propose an edge plausibility metric based on graph embedding that enables us to exploit this weakness in order to identify fake edges. Extensive experiments show that, using this metric, we are able to recover the original graph from an anonymized graph to a large extent. Our graph recovery also results in significant privacy damage to the original anonymization mechanisms. To mitigate this weakness, we propose enhancement over the existing anonymization mechanisms. Our experiments show that our enhanced mechanisms significantly reduce the performance of our graph recovery attack, increase graph de-anonymization resistance, and at the same time provide better graph utility.

ACKNOWLEDGMENT

REFERENCES

[1] L. A. Adamic and E. Adar, "Friends and Neighbors on the Web," *Social Networks*, 2003. 5

[2] L. M. Aiello and N. Barbieri, "Evolution of Ego-networks in Social Media with Link Recommendations," in *Proceedings of the 2017 ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, 2017, pp. 111–120. 4

[3] M. Backes, P. Berrang, A. Hecksteden, M. Humbert, A. Keller, and T. Meyer, "Privacy in Epigenetics: Temporal Linkability of MicroRNA Expression Profiles," in *Proceedings of the 2016 USENIX Security Symposium (USENIX Security)*.   USENIX, 2016, pp. 1223–1240. 6

[4] M. Backes, M. Humbert, J. Pang, and Y. Zhang, "walk2friends: Inferring Social Links from Mobility Profiles," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2017, pp. 1943–1957. 1, 4, 5, 6

[5] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography," in *Proceedings of the 2007 International Conference on World Wide Web (WWW)*.   ACM, 2007, pp. 181–190. 1, 13

[6] L. Backstrom and J. Leskovec, "Supervised Random Walks: Predicting and Recommending Links in Social Networks," in *Proceedings of the 2011 ACM International Conference on Web Search and Data Mining (WSDM)*.   ACM, 2011, pp. 635–644. 4

[7] G. Beigi and H. Liu, "Privacy in Social Media: Identification, Mitigation and Applications," *CoRR abs/1808.02191*, 2018. 1

[8] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava, "Class-based Graph Anonymization for Social Network Data," *Proceedings of the VLDB Endowment*, 2009. 13

[9] J. Cheng, A. W. chee Fu, and J. Liu, "K-Isomorphism: Privacy Preserving Network Publication against Structural Attacks," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD)*. ACM, 2010, pp. 459–470. 13

[10] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and Mobility: User Movement in Location-based Social Networks," in *Proceedings of the 2011 ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2011, pp. 1082–1090. 1

[11] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends in Theoretical Computer Science*, 2014. 1, 10

[12] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010. 12

[13] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing," in *Proceedings of the 2014 USENIX Security Symposium (USENIX Security)*. USENIX, 2014, pp. 17–32. 6

[14] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *Proceedings of the 2016 ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2016, pp. 855–864. 1, 2, 4, 5

[15] I. Hagestedt, Y. Zhang, M. Humbert, P. Berrang, H. Tang, X. Wang, and M. Backes, "MBeacon: Privacy-Preserving Beacons for DNA Methylation Data," in *Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2019. 6

[16] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Proceedings of the 2017 Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2017, pp. 1025–1035. 4

[17] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting Structural Re-identification in Anonymized Social Networks," *Proceedings of the VLDB Endowment*, 2008. 13

[18] S. Ji, W. Li, P. Mittal, X. Hu, and R. Beyah, "SecGraph: A Uniform and Open-source Evaluation System for Graph Data Anonymization and De-anonymization," in *Proceedings of the 2015 USENIX Security Symposium (USENIX Security)*. USENIX, 2015, pp. 303–318. 1, 3, 5, 6, 10, 12

[19] S. Ji, W. Li, M. Srivatsa, and R. Beyah, "Structural Data De-anonymization: Quantification, Practice, and Implications," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2014, pp. 1040–1053. 13

[20] S. Ji, P. Mittal, and R. Beyah, "Graph Data Anonymization, De-Anonymization Attacks, and De-Anonymizability Quantification: A Survey," *IEEE Communications Surveys & Tutorials*, 2016. 1

[21] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2019. 6

[22] J. Jia, B. Wang, L. Zhang, and N. Z. Gong, "AttriInfer: Inferring User Attributes in Online Social Networks Using Markov Random Fields," in *Proceedings of the 2017 International Conference on World Wide Web (WWW)*. ACM, 2017, pp. 1561–1569. 1

[23] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the Spread of Influence through a Social Network," in *Proceedings of the 2003 ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2003, pp. 137–146. 1

[24] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*. Cambridge University Press, 2014. 12

[25] D. Liben-Nowell and J. Kleinberg, "The Link-prediction Problem for Social Networks," *Journal of the American Society for Information Science and Technology*, 2007. 1, 2, 4

[26] K. Liu and E. Terzi, "Towards Identity Anonymization on Graphs," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD)*. ACM, 2008, pp. 93–106. 1, 2, 3, 13

[27] J. McAuley and J. Leskovec, "Learning to Discover Social Circles in Ego Networks," in *Proceedings of the 2012 Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2012, pp. 539–547. 5

[28] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. Vadhan, "The Limits of Two-Party Differential Privacy," in *Proceedings of the 2010 Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2010, pp. 81–90. 9

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proceedings of the 2013 International Conference on Learning Representations (ICLR)*, 2013. 4

[30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionally," in *Proceedings of the 2013 Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2013, pp. 3111–3119. 4

[31] P. Mittal, C. Papamanthou, and D. Song, "Preserving Link Privacy in Social Network Based Systems," in *Proceedings of the 2013 Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2013. 1, 3, 13

[32] A. Narayanan and V. Shmatikov, "De-anonymizing Social Networks," in *Proceedings of the 2009 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2009, pp. 173–187. 2, 10, 12, 13

[33] S. Nilizadeh, A. Kapadia, and Y.-Y. Ahn, "Community-enhanced De-anonymization of Online Social Networks," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2014, pp. 537–548. 13

[34] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási, "Structure and Tie Strengths in Mobile Communication Networks," *Proceedings of the National Academy of Sciences*, 2007. 1

[35] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online Learning of Social Representations," in *Proceedings of the 2014 ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2014, pp. 701–710. 1, 2, 4, 5

[36] D. Proserpio, S. Goldberg, and F. McSherry, "A Workflow for Differentially-private Graph Synthesis," in *Proceedings of the 2012 ACM Workshop on Online social networks (WOSN)*. ACM, 2012, pp. 13–18. 13

[37] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro, "Knock Knock, Who's There? Membership Inference on Aggregate Location Data," in *Proceedings of the 2018 Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2018. 6

[38] T. Rahman, B. Surma, M. Backes, and Y. Zhang, "Fairwalk: Towards Fair Graph Embedding," in *Proceedings of the 2019 International Joint Conferences on Artifical Intelligence (IJCAI)*. IJCAI, 2019, pp. 3289–3295. 4

[39] D. M. Romero, B. Meeder, and J. Kleinberg, "Differences in the Mechanics of Information Diffusion Across Topics: Idioms, Political Hashtags, and Complex Contagion on Twitter," in *Proceedings of the 2011 International Conference on World Wide Web (WWW)*. ACM, 2011, pp. 695–704. 1

[40] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao, "Sharing Graphs using Differentially Private Graph Models," in *Proceedings of the 2011 ACM Internet Measurement Conference (IMC)*. ACM, 2011, pp. 81–98. 1, 3, 6

[41] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models," in *Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2019. 6

[42] K. Sharad and G. Danezis, "An Automated Social Graph De-anonymization Technique," in *Proceedings of the 2014 Workshop on Privacy in the Electronic Society (WPES)*. ACM, 2014, pp. 47–58. 13

[43] M. Srivatsa and M. Hicks, "Deanonymizing Mobility Traces: Using Social Network as a Side-channel," in *Proceedings of the 2012 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2012, pp. 628–637. 13

[44] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002. 1

[45] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale Information Network Embedding," in *Proceedings of the 2015 International Conference on World Wide Web (WWW)*. ACM, 2015, pp. 1067–1077. 4

[46] B. Thompson and D. Yao, "The Union-Split Algorithm and Cluster-Based Anonymization of Social Networks," in *Proceedings of the 2009 ACM Asia Conference on Computer and Communications Security (ASIACCS)*. ACM, 2009, pp. 218–227. 13

[47] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the Evolution of User Interaction in Facebook," in *Proceedings of the 2009 ACM Workshop on Online social networks (WOSN)*. ACM, 2009, pp. 37–42. 2, 5

[48] Y. Wang and X. Wu, "Preserving Differential Privacy in Degree-correlation based Graph Generation," *Transactions on Data Privacy*, 2013. 13

[49] Q. Xiao, R. Chen, and K.-L. Tan, "Differentially Private Network Data Release via Structural Inference," in *Proceedings of the 2014 ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2014, pp. 911–920. 1, 3, 13

[50] Y. Zhang, "Language in Our Time: An Empirical Analysis of Hashtags," in *Proceedings of the 2019 Web Conference (WWW)*. ACM, 2019, pp. 2378–2389. 1

[51] Y. Zhang, M. Humbert, T. Rahman, C.-T. Li, J. Pang, and M. Backes, "Tagvisor: A Privacy Advisor for Sharing Hashtags," in *Proceedings of the 2018 Web Conference (WWW)*. ACM, 2018, pp. 287–296. 1

[52] B. Zhou and J. Pei, "Preserving Privacy in Social Networks Against Neighborhood Attacks," in *Proceedings of the 2008 International Conference on Data Engineering (ICDE)*. IEEE, 2008, pp. 506–515. 3, 13

[53] L. Zou, L. Chen, and M. T. Ozsu, "K-Automorphism: A General Framework for Privacy Preserving Network Publication," *Proceedings of the VLDB Endowment*, 2009. 13