This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

Disclose More and Risk Less: Privacy Preserving Online Social Network Data Sharing

Jiayi Chen, Jianping He, *Member, IEEE,* Lin Cai, *Senior Member, IEEE,* and Jianping Pan, *Senior Member, IEEE*

Abstract—Many third-party services and applications have integrated the login services of popular Online Social Networks, such as Facebook and Google+, and acquired user information to enrich their services by requesting user's permission. Although users can control the information disclosed to the third parties in a certain granularity, there are still serious privacy risks due to the inference attack. Even if users conceal their sensitive information, attackers can infer their secrets by exploiting the correlations among private and public information with background knowledge. To defend against such attacks, we formulate the social network data sharing problem through an optimization-based approach, which maximizes the users' self-disclosure utility while preserving their privacy. We propose two privacy-preserving social network data sharing methods to counter the inference attack. One is the efficiency-based privacy-preserving disclosure algorithm (EPPD) targeting the high utility, and the other is to convert the original problem into a multi-dimensional knapsack problem (d-KP) using greedy heuristics with a low computational complexity. We use real-world social network datasets to evaluate the performance. From the results, the proposed methods achieve a better performance when compared with the existing ones.

Index Terms—Inference attack, Online social network, Privacy, Data sharing.

1 Introduction

Online Social Networks (OSNs) have attracted an increasing number of people to build their social relations on the Internet. The accounts in popular social network sites, such as Facebook and Google+, have become people's second identity since they record users' detailed profile information (attributes) and interpersonal relationships (social relations). People are willing to share part of their personal information to find new friends with similar interests, which is called *self-disclosure* [1]. Due to the *privacy concerns* [2], OSN users are reluctant to disclose their full set of personal information. Therefore, the social network service providers allow users to determine whether a specific field is open to the public or not. However, it is still possible to infer the hidden and secret information of OSN users with high accuracy by exploiting the public information. As a result, unsafe self-disclosure may be followed by potential privacy leaks, leading to targeted spams, reputation damage, and even property loss.

Nowadays, the third-party applications can access user profiles and relationships with user's authorization, so they can leverage the application-specific social networks and integrate their services with the existing OSNs. OAuth 2.0 [3], a common authorization protocol, has been designed to guarantee the authorization security and simplify the authorization process without sharing users' credentials. The authorization mechanism allows each user to know the resources that the third-party applications require and then to determine whether to grant the resource access permission. However, even if users have a full control of what to disclose, they have little knowledge of whether the third-party applications can exploit the disclosed information to infer their secrets. Meanwhile, the user experience with the OSNs and applications depend on safe self-disclosure. It is critical to ensure that users enjoy the benefits of the services free from the worries about privacy leakage.

Different from the privacy-preserving anonymized data publishing, social network data provided to the third parties are sometimes required to be only partially or even not anonymized (e.g., logging services with Facebook account information). The main privacy concern here is the inference attack on user secrets. For various types of inference methods [4], [5], [6], [7], the principle is to find out the targeted secrets based on the information extracted from the published dataset and background knowledge of the attackers. For example, an attacker can train a classifier from the training dataset to predict whether a user has a certain secret. When a new social network is published, the attacker extracts features from the public information of the targeted user as the input of the classifier and then infers the secret. In the whole process, the training dataset can be viewed as the background knowledge, and the extracted features can be regarded as the observation. Both the quality of the training set and the performance of the attacker's classifier

- *J. Chen, L. Cai are with the Department of Electrical and Computer Engineering, Unviersity of Victoria, Victoria, BC, V8P 5C2, Canada. E-mail: jychen21@uvic.ca; cai@ece.uvic.ca.*
- *J. He is with the Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, China. E-mail: jphe@sjtu.edu.cn.*
- *J. Pan is with the Department of Computer Science, University of Victoria, Victoria, BC, V8P 5C2, Canada. E-mail: pan@uvic.ca.*

- *Manuscript received October 18, 2017; accepted July 19, 2018. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), Canada Foundation for Innovation (CFI), BC Knowledge Development Fund (BCKDF), and the Natural Science Foundation of China (NSFC) under grant 61773257, 61761136012.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

affect the effectiveness of the inference attack.

A direct way of defending against the inference attack is to pre-process the released network data. There are several common operations to reduce the chance of inference attacks, such as *masking* (removing data), *obfuscation* (adding confusing data), and *generalization* (coarsening data) [8]. In this work, we prefer to mask the secret-related information rather than to add misleading information to maintain the data utility because misleading or fake information may bring users other troubles (e.g., inaccurate news feeds and friend recommendations) in non-anonymized social networks. According to the survey [9] on privacy protection strategies on Facebook, the interviewed students were strongly against using fake information as a privacy protection method because of the confusion brought by the fake information. However, the excessive concealment of user attributes will reduce the self-disclosure utility of OSN users, which results in the degradation of user experience and application performance. Thus, we need to mask attributes effectively and efficiently with the consideration of both the self-disclosure utility and privacy concerns.

Therefore, our goal is to design the algorithms for releasing as much social network data as possible while satisfying the privacy guarantees according to different user concerns. To achieve the goal, there are three main challenges. i) Privacy Concerns: Different users may have different privacy concerns. A certain profile attribute may be a secret to one person while it may not be sensitive to another; ii) Attacker's Ability: It is necessary to design a general protection algorithm with regard to attackers' ability. But the background knowledge and capabilities of an attacker are usually unknown; and iii) Privacy and Utility Evaluation: To establish the privacy protection model, we need to quantify, evaluate and make a trade-off between the extent of self-disclosure and privacy leakage.

In this paper, we first formulate the privacy-preserving online social network data sharing problem as a knapsack-like problem, and then propose two social network data disclosure methods, an efficiency-based privacy-preserving disclosure (EPPD) algorithm and a multi-dimensional knapsack problem (d-KP) simplification based method, respectively. The main contributions of this paper are threefold.

- We use the social-attribute network model to describe both the social network data and attacker's knowledge, and propose the self-disclosure rate to quantify the leakage of user secret in the published network regardless of the attacker's knowledge.
- To defend the inference attack, we formulate a novel privacy-preserving social network data sharing problem, which maximizes the user self-disclosure utility with privacy guarantees. The optimization problem also takes different user concerns into account and enables a flexible self-disclosure evaluation in order to satisfy different user demands and scenarios.
- The two proposed social network data sharing methods are designed for different purposes and protection levels. The EPPD algorithm targets the high utility satisfying the formal privacy protection constraints, while the d-KP disclosure algorithm greatly reduces the computational complexity of solving the social relation disclosure problem. We use two real-world social networks to evaluate the performance and compare them with the existing work.

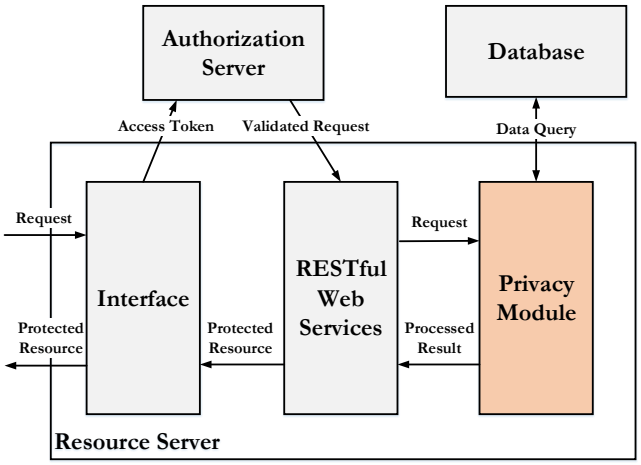


Fig. 1: A simple privacy protection framework based on OAuth 2.0 protocol.

The remainder of this paper is organized as follows. Section 2 introduces some important preliminaries for the problem formulation including modeling of social-attribute networks, inference attack model, adversarial abilities, and the definition of utility. Section 3 defines the privacy disclosure and formulates the problem, followed by the privacy protection methods proposed in Section 4. In Section 5, we conduct several experiments with real-world datasets and verify the efficiency of the proposed methods. In Section 6, we introduce the related work. Finally, we conclude the paper in Section 7 with future work.

2 Preliminaries

2.1 Scenario

In our scenario, the main concern is the potential privacy leak led by the inference attack when the OSN service provider (host) releases a user's information to the third-party applications. We assume that the host OSN service provider is fully trusted by the user, while the third-party applications are not. On one hand, the user only authorizes the third-party applications to access a number of limited resources. On the other hand, some third-party applications may be curious about the user's secrets, and try to infer them based on the available information and their background knowledge. The host is responsible for controlling the information to release and ensure that the disclosed information is insufficient for the third-party applications to infer the secrets effectively. Thus, the OSN data disclosure algorithms should be implemented in the servers of the OSN service providers.

Figure 1 is an example of implementation based on the OAuth 2.0 framework [3] with the "Privacy Module" as the critical part to protect users' privacy. OAuth 2.0 defines four roles, including client (third-party applications), resource owner (OSN users), authorization server and resource server (the host OSN service providers). To be compatible with the existing framework, the privacy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

TABLE 1: Notation

Symbol	Definition
V_N	Vertex set of social actors
V_A	Vertex set of attributes
E_N	Edge set of social relations
E_A	Edge set of attribute links
$\mathcal{N}_u$ / $\mathcal{N}_a$	Social actors connected with actor u/attribute a
$\mathcal{A}_u$	Attribute set of actor u
$\mathcal{S}_u$	Secret attributes of actor u
$\mathcal{P}_u$	Public attributes of actor u
G_A	Attack graph
G_P	Published social network graph
$t_u(s)$	Indicating whether actor u has secret s
$\Phi(u, s, G_P)$	Self privacy disclosure of actor u's secret s in G_P
p_i	Value of the ith attribute or social relation
$\epsilon, \mathcal{E}$	Privacy budget (set)
δ, Δ	Relaxation variable (set)
θ, Θ	Privacy threshold (set) determined by ϵ and δ

module with the data disclosure algorithms is implemented in the resource server, which intercepts the data access (mainly the READ operation) to the database storing users' personal information. As shown in Fig. 1, once the third-party applications send the authorized request with a valid access token to the resource server, the privacy module will acquire the requester, the targeted user, the requested fields, and some other additional information, and then decide what to disclose as the protected resources to the third-party applications.

In the next part, we will introduce some preliminaries including the social network model, and the inference attack as well as the utility of the third-party services before formulating the privacy-preserving data sharing problem in OSNs. Table 1 shows the basic notations commonly used in our work.

2.2 Social-Attribute Network

A social network is usually described as a graph consisting of social actors and relationships where user attributes (e.g., profile information) are used to label or group social actors. To present both social actors and attributes together, we use a social-attribute network graph $G = \{V_N, V_A, E_N, E_A\}$, where V_N is the social actor set, V_A is the attribute node set, E_N is the social relation set and E_A is the attribute link set. The social-attribute network model is first proposed by Yin et al. [10], and it is widely used for social network analysis, link prediction and hidden attribute inference [11], [12]. The social network model used in Fig. 2 is a social-attribute network. There are two types of attributes: *categorical attribute* and *numerical attribute*. A categorical attribute belongs to a certain category in the user profile, where all candidates can be enumerated. If an attribute is described as a number, it can be regarded as a numerical attribute.

In our model, in order to represent a numerical attribute with a node, it has to be converted into a categorical attribute by using the interval or ordinal variables. For example, "Age: 26" can be shown as "Age: 20–29" or "Young". For simplicity, despite the fact that a category can be regarded as a tree with attributes at different granularities (e.g., location can be "Mountain View, CA" or only "CA", and the former attribute is the child node of the latter one), we consider all attributes belonging to the same category are in the same level of the tree.

A social relation $(u, v) \in E_N$ means that u and v are friends in an undirected network, or u follows v in a directed network, while an attribute link $(u, v) \in E_A$ means that u has the attribute v. We use an indicator function $t_u^G(v) \in \{0, 1\}$ to indicate the existence of edge (u, v) in graph G. In a social-attribute network, there are two kinds of edges: *actor-to-attribute* and *actor-to-actor*. We can use the neighborhood information to form node sets where nodes have the same attributes or common friends. For a social actor u, the friend set (in undirected networks) of node u is denoted as $\mathcal{N}_u = \{v | v \in V_N, (u, v) \in E_N\}$, and the attribute set of social actor u is denoted as $\mathcal{A}_u = \{a | a \in V_A, (u, a) \in E_A\}$. Similarly, social actors sharing the same attribute a are all involved in the set $\mathcal{N}_a = \{v | v \in V_N, (v, a) \in E_A\}$. Furthermore, we can obtain the social actor set with multiple common attributes and relationships by calculating the intersection of the corresponding neighborhood sets. For example, A's friends who are photographers can be expressed by $\mathcal{N}_A \cap \mathcal{N}_{\text{Photographer}}$. For convenience, we introduce some other symbols. The attributes of the node u can be divided into two sets, the secret attributes $\mathcal{S}_u$ and the public attributes $\mathcal{P}_u$, where $\mathcal{A}_u = \mathcal{S}_u \cup \mathcal{P}_u, \mathcal{S}_u \cap \mathcal{P}_u = \emptyset$.

2.3 Privacy Inference Attack

In a social-attribute network, the privacy inference attack (sensitive attributes and relations inference) can be regarded as a special form of link prediction [12]. An attacker exploits both the public information from published social networks and the background knowledge to infer user privacy. Since the attacker can adopt a variety of techniques with different background knowledge, it is difficult to take all situations into consideration. However, it is feasible to model an attacker into a knowledge graph [13], [14], [15]. Similarly, we use another social-attribute network graph G_A (attack graph) to describe the background knowledge of an attacker. An attack graph can contain the following information:

1) *Statistical Information*. It can be obtained from official statistics directly or from the published dataset. A piece of statistical information can be expressed as a conditional probability of owning a secret given several attributes.
2) *Node & Edge Information*. An attack graph can contain node and edge information not contained in the published social networks. It can involve part of the original social network, and additional edges from other social networks as well.

Different from the social-attribute network for the original dataset, an attack graph translates each piece of statistical information into a tree-like inference path: *Attributes-Feature-Secret*, where *Feature* node is treated as a virtual social actor with several *Attribute* nodes connecting to it and a weighted edge to a *Secret* node. The weight is the conditional probability $\Pr(\text{Secret}|\text{Attributes})$. For example, the statement "*People with attributes* $A1$ *and* $A2$ *have a probability of 90% to own secret* $S1$" can be expressed as path

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

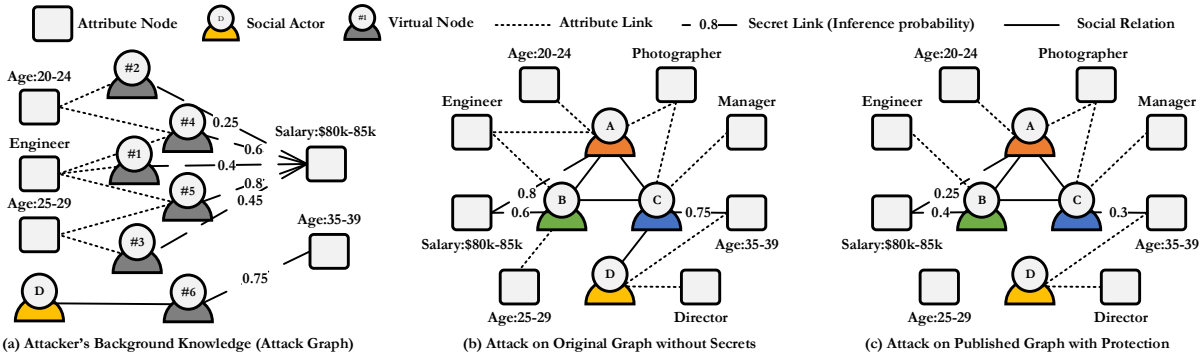


Fig. 2: An example of the attribute inference attack in OSNs. According to #4 and #5 in the attack graph (a), the attacker infers that A and B in the original graph (b) have a high probability of earning $80k–85k annually. According to #6, the attacker knows that around 75% friends of D are in the same age group, which implies that C, as D's friend, is probably around 35 to 39 years old. However, attackers can hardly make accurate inferences on processed graph (c) since only 25% of people in the age group of 20–24 and 40% of engineers have the salary of $80k–85k. Also, for C, after removing the relation between C and D, it is hard to guess C's true age from other attributes.

A1,A2-F1-S1 where $F1$ is a virtual node representing social actors with attribute $A1$ and $A2$. Fig. 2 gives a toy example of the inference attack and a feasible defending method against it. With sufficient information observed from the social-attribute network, an adversary can easily conduct the inference attack based on the background knowledge. After removing a few attributes and social relations, the probability of inferring the secrets of the targeted OSN user will greatly decrease.

2.4 Adversarial Ability

As mentioned in the scenario, the third party is considered as the potential attacker who is curious about the users' hidden information. After a user provides the third-party service with public information stored in the host social network service, the third party can exploit this information to generate a formatted input for its attack model whose output is whether the user has a secret attribute. The attack model can be a well-trained classifier or a statistical model. To obtain such a model, the attacker can use the data from its own database or the public anonymized data collections. As a result, both the host social network and the user can hardly predict the adversarial ability. Therefore, to achieve a good protection performance, it is necessary to assume that the adversary has a strong attack capability.

In this paper, we assume that the adversary has almost the same dataset as the host social network service, which means that the adversary has the identical topology of the social-attribute network as the original one. In addition, we make some specific assumptions for the privacy inference attack via public attributes and social relations, respectively. For the former one, we assume that the adversary has the "anonymized" version of the original graph so that it can obtain full statistical information about public attributes and secrets. For the latter one, we assume that the adversary knows the exact number of common neighbors who have the secret attributes with the given set of users. In Section 3, we introduce the self privacy disclosure based on the adversarial ability.

2.5 Utility

To defend against the inference attack and guarantee users' privacy, it is essential to limit the information to disclose, which can cause the loss of utility. The third parties are always expecting as much information from users as possible because it not only helps them improve services (e.g., providing more accurate recommendation), but also provides useful information for marketing and analytics. However, sharing too much social network data exposes users into the threats of privacy inference attack. Therefore, there is always a trade-off between the utility and privacy.

Before we give the privacy definition for the user side, we also need to quantify the utility for the third parties. In this paper, We regard the utility as the value of an attribute or a social relation which is denoted as p, and to determine the utility of disclosed information T, we have a utility function $p(T)$. The most intuitive and common way of determining the utility is to consider every piece of information equally ($p = 1$) and get the sum of them [16]. Nevertheless, we provide some other utility options for both attribute utility and relation utility.

2.5.1 Attribute Utility

The utility of an attribute highly depends on its semantic properties and the requirements of the third parties. For example, a location attribute "Mountain View, CA" contains more information than "CA" alone, and the location-based services usually prefer the detailed location attributes to the others. Therefore, it is difficult to find a uniform standard to evaluate the attribute. To measure the attribute utility in a general way, we focus on the neighbors of the attribute in the social-attribute network, which naturally shows how common the attribute is in the whole social network.

Uniqueness: the information carried by an attribute can be measured as $-\log(1/|\mathcal{N}_a|)$ according to the information theory. Intuitively, an attribute with fewer social actor neighbors is more "informative", and makes a user more "unique". Here, we use the inverse of the attribute node's

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

degree centrality to calculate the uniqueness score.

$$p_U(a) = \frac{1}{\log(|\mathcal{N}_a|) + 1}. \quad (1)$$

Commonness: sometimes, the third parties are more interested in the common attributes for analysis. We can simply use the normalized degree centrality in the whole network. To ensure the structure of a community, we use the degree centrality in the ego network of the targeted social actor.

$$p_C(a, u) = \frac{|\mathcal{N}_u \cap \mathcal{N}_a|}{|\mathcal{N}_u|}. \quad (2)$$

2.5.2 Social Relation Utility

An edge's value can be determined by the node similarity between two social actors connected. Intuitively, two similar social actors (sharing a lot of attributes in common) have a stronger tie between them.

Jaccard Coefficient: The normalized common neighbor metric describes the similarity of two social actors. The more common friends will bring a higher Jaccard coefficient.

$$p_J(e_{u,v}) = \frac{|\mathcal{N}_u \cap \mathcal{N}_v|}{|\mathcal{N}_u \cup \mathcal{N}_v|}. \quad (3)$$

Adamic/Adar Score: Adamic et al. [17] proposed this score to describe the similarity between two web pages. Different from the Jaccard coefficient, it considers the weight of each common feature. For our experiments, Adamic/Adar score is defined as follows.

$$p_A(e_{u,v}) = \sum_{k \in \mathcal{F}_u \cap \mathcal{F}_v} \frac{1}{\log |\mathcal{N}_k|}, \quad (4)$$

where the common feature with a smaller degree centrality weighs more.

3 Problem Formulation

3.1 Self Privacy Disclosure

To quantify the privacy disclosure, Martin et al. [18] defined the disclosure risk as the likelihood of the most possible sensitive attribute assignment with respect to the background knowledge for privacy-preserving data sharing. In the case of social networks, the disclosure risk of a binary secret s of the social actor u in a published network graph G_P given a certain background knowledge graph G_A is:

$$\Pr\{t_u(s) = 1 | G_A, G_P\} = \Pr\{t_u^A(s) = 1 | g(\mathcal{A}_u^P)\}, \quad (5)$$

where the function $\mathcal{A}_u^A = g(\mathcal{A}_u^P)$ maps the attribute set $\mathcal{A}_u^P$ onto $\mathcal{A}_u^A$ in the attack graph G_A.

However, since the background knowledge varies from attackers to attackers, we should consider a general privacy measurement without regard to a certain attack graph. Therefore, we introduce the *self privacy disclosure* to evaluate the disclosure rate of a certain social actor n's secret s in the graph G.

Based on the adversarial ability mentioned in the previous section, we can further define the self privacy disclosure under a strong attacker. The self privacy disclosure from the perspective of attributes, $\Phi_{\mathcal{A}}$, can be calculated by

$$\Phi_{\mathcal{A}}(u, s, G_P) \triangleq \Pr\{t_u(s) = 1 | \mathcal{A}_u \cap \mathcal{A}_u^P\}. \quad (6)$$

Also, we can process the social relations in a similar way to attributes. Let $\Phi_{\mathcal{N}}$ be the self privacy disclosure which is the likelihood of owning secret s given the $\mathcal{N}_u$.

$$\Phi_{\mathcal{N}}(u, s, G_P) \triangleq \Pr\{t_u(s) = 1 | \mathcal{N}_u \cap \mathcal{N}_u^P\}. \quad (7)$$

Similarly to the existing definitions of privacy such as Bayes-optimal privacy [19], indistinguishable privacy [20] and differential privacy [21], [22], we define the threshold for self privacy disclosure as the privacy guarantee. An operation is considered to be privacy-preserving if it satisfies the constraint that the difference between the attacker's prior and posterior beliefs about the sensitive information is small enough, which can be expressed as follows.

$$\Phi(u, s, G_P) \leq \exp(\epsilon) \Pr\{t_u(s) = 1\} + \delta, \quad (8)$$

where $\Pr\{t_u(s) = 1\}$ is the prior probability of u's secret s. Here, we use two parameters, ϵ and δ, to control the privacy protection strength. ϵ is the privacy budget, a non-negative parameter to define how close the self privacy disclosure rate is to the prior probability. It determines the strict privacy threshold at $\delta = 0$, and therefore, it is usually set to a small number. δ controls the tolerance of privacy disclosure, and it should be numerically smaller than $1 - \exp(\epsilon) \Pr\{t_u(s) = 1\}$ since the range of self privacy disclosure is $[0, 1]$. δ enables a flexible way to relax the privacy constraint which allows to customize different privacy requirements for different situations.

In this paper, we define a user's privacy concern as a set of tuples consisting of secret attributes and their privacy requirements. Each tuple (s, ϵ, δ) is representing a piece of privacy setting that follows the privacy definition shown in (8). When we take all users' privacy concern into account, and denote it as $C = \{\mathcal{S}, \mathcal{E}, \Delta\}$, which is the aggregation of all tuples, the disclosed graph G_P can be regarded privacy-preserving if it satisfies

$$\Phi(u, \mathcal{S}_u, G_P) \leq \Theta_u, \forall u \in V_N, \quad (9)$$

where Θ_u is the vector of privacy thresholds $\theta_{u,i}$ which can be calculated as

$$\theta_{u,i} = \exp(\epsilon_{u,i}) \Pr\{t_u(S_{u,i}) = 1\} + \delta_{u,i}, \quad (10)$$

where $\epsilon_{u,i} \in \mathcal{E}, \delta_{u,i} \in \Delta, i = 1, 2, \ldots, |S_u|$.

Note that since the disclosed graph G_P is a subgraph of the original graph which is supposed to remove part of the edges (i.e., social relations and attribute links), we can also use the disclosed edges $T \subseteq E_N \cup E_A$ to represent the disclosed graph.

3.2 General Form

The goal of our work is to mask part of edges in the social-attribute network so that users can disclose as much valuable personal information as possible with privacy guarantees. The masking operations start with all edges involved and then remove one edge after another from the network. Here we process the network in the opposite way: at the beginning, the network includes all the nodes only, both social actors and attributes, and we add edges into it. Then, the general social network data sharing problem is formulated as follows.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

Given a social network graph $G = \{V_N, V_A, E_N, E_A\}$, all users' privacy concern $C = \{\mathcal{S}, \mathcal{E}, \Delta\}$ and the utility function $p(T)$, obtain a disclosed social network G_p with the disclosed edge set $T = E'_N \cup E'_A$, where $E'_N \subseteq E_N$ and $E'_A \subseteq E_A$ such that the disclosed social network satisfies privacy requirements given in (9) with the maximum utility. And the maximum utility can be obtained by

$$y = \max_{T \subseteq E_N \cup E_A} \{p(T) : \Phi(u, S_u, T) \leq \Theta_u, \forall u \in V_N\}, \quad (11)$$

where Θ_u can be calculated with $\mathcal{S}$, $\mathcal{E}$, and Δ by (10) .

However, the self privacy disclosure function does not hold either submodularity or monotonicity. This can be shown by the following proofs.

Non-submodular: Given the social network G with $V_N = \{a, b, c, d, e, f\}$, $V_A = \{S, A_1, A_2\}$, and $E_A = \{(a, S), (a, A_1), (a, A_2), (b, S), (b, A_1), (c, S), (c, A_1), (d, A_1), (d, A_2), (e, A_1), (f, A_2)\}$ where S is the secret of user a, there exist $e = (a, A_2)$, $T_1 = E_A \setminus \{(a, S), (a, A_1), (a, A_2)\}$ and $T_2 = E_A \setminus \{(a, S), (a, A_1)\}$, $T_1 \subseteq T_2$ such that

$$g(T_1 \cup \{e\}) - g(T_1) < g(T_2 \cup \{e\}) - g(T_2), \quad (12)$$

where $g(T) = \Phi(a, S, T)$. Since the example here does not follow the definition of the submodular function, we can conclude that the self privacy disclosure function is non-submodular with the fixed u and S_u. □

Non-monotonic: Given the same conditions as the previous proof, there exist $e_1 = (a, A_1)$, $e_2 = (a, A_2)$ and $T = E_A \setminus \{(a, S), (a, A_1), (a, A_2)\}$ such that

$$g(T) < g(T \cup \{e_1\}), g(T \cup \{e_1\}) > g(T \cup \{e_1, e_2\}), \quad (13)$$

which violates the definition of monotonicity. Therefore, the self privacy disclosure function is non-monotonic with the fixed u and S_u. □

Due to the properties of the self privacy disclosure function, it is difficult to solve the general social network data sharing problem directly. Therefore, we need to further process the general problem. Because of the different assumptions on adversarial abilities for public attributes and social relations, we separately consider attribute and relation disclosure problems derived from the general form for simplification. In Section 4, we give two privacy protection algorithms to obtain the feasible solutions with low computational complexity, where one of the proposed algorithms, d-KP simplification, is designed to further simplify the problem into a well-studied one.

3.3 Attribute Disclosure Problem

According to our assumptions on the attacker' ability, each user has an independent profile which means that the change of a user's profile will not affect the other users' disclosure strategies. Therefore, we can process the attribute disclosure problem for each user. Then, we introduce an optimization problem as follows.

$$\max_{\mathbf{x}} \quad \sum_{i=1}^{|\mathcal{P}_u|} p_i x_i \quad (14)$$

$$\text{s.t.} \quad \Phi_{\mathcal{A}}(u, s_j, \mathbf{x}) \leq \theta_j, \qquad j = 1, \ldots, |\mathcal{S}_u|, \quad (15)$$

$$x_i \in \{0, 1\}, \qquad i = 1, \ldots, |\mathcal{P}_u|, \quad (16)$$

where each element x_i in vector $\mathbf{x}$ indicates whether to disclose the corresponding public attribute a_i with utility value p_i which can be defined as the value of the attribute for different purposes (see Section 2.5). $\Phi_{\mathcal{A}}(u, s_j, \mathbf{x})$ is the self privacy disclosure of node u's secret s_j according to vector $\mathbf{x}$, which is defined in (6). θ_j is the privacy protection threshold for secret s_j, which is equal to $\exp(\epsilon) \Pr\{t_u(s_j) = 1\} + \delta_j$. Note that the disclosed social network here is the subgraph of the original network. The self privacy disclosure is actually determined by selected attributes, and thus, we use $\Phi_{\mathcal{A}}(u, s_j, \mathbf{x})$ instead of $\Phi_{\mathcal{A}}(u, s_j, G_P)$.

3.4 Relation Disclosure Problem

As described above, the attribute disclosure problem for each social actor can be solved locally without the involvement of other social actors. However, a social relation involves two actors, and therefore, when we disclose a relation for a social actor, we should take into account the influence of this relation on the other actor. Hereby, we formulate the social relation disclosure problem as follows.

In the directed social networks, we mainly consider the successors of a social actor since users can determine who to follow, but not their followers. Besides, the removal of a directed social relation does not affect its reverse relation. Therefore, we can process the directed social relation disclosure problem in a similar way to the attribute disclosure problem by changing $\mathcal{P}_u$ into $\mathcal{N}_u$ and using the $\Phi_{\mathcal{N}}(u, s_j, G_P)$ as the constraints.

In the undirected social networks, the relation disclosure problem cannot be solved for each social actor, because the removal of one undirected edge will affect two social actors. Therefore, we need to address the relation disclosure problem as a whole, which is formulated as follows.

$$\max_{\mathbf{x}} \quad \sum_{i=1}^{|E_N|} p_i x_i \quad (17)$$

$$\text{s.t.} \quad \Phi_{\mathcal{N}}(u_k, s_{k,j}, \mathbf{x}) \leq \theta_{k,j}, \quad j = 1, \ldots, |\mathcal{S}_{u_k}|, k = 1, \ldots, |V_N|, \quad (18)$$

$$x_i \in \{0, 1\}, \ i = 1, \ldots, |E_N|, \quad (19)$$

where $\theta_{k,j} = \exp(\epsilon) \Pr\{t_{u_k}(s_{k,j}) = 1\} + \delta_{k,j}$ and all protection constraints including all social actors and their secrets are considered together. Compared with the optimization problem formulated in (14), the main challenge is the complexity of the undirected social relation disclosure problem. Thus, to address this problem, we specifically propose a d-KP simplification method, which will be introduced in detail in the next section.

4 Privacy Protection Algorithms

4.1 Overview

Our social network publishing problem is a knapsack-like combinatorial optimization problem. For each attribute or relation, we can regard the utility (e.g., self-disclosure) as its "profit" or "value", the contribution to secret disclosure as its "weight", and the privacy protection threshold as the "maximum weight". For the knapsack problem, there exists an exact solution by using dynamic programming (DP) or Branch and Bound [23]. Similarly, we can use DP to obtain

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

a feasible solution to our proposed problem. The maximum value of selected attributes or social relations can be defined as the following general form:

$$p_{\max}(i,C) = \begin{cases} 0, & i \leq 1, i > n, \\ p_{\max}(i-1,C), & i > 1, \exists w > \theta, \\ \max(p_{\max}(i-1, C \cap N_i) + p_i, & \\ \quad p_{\max}(i-1,C)), & \text{otherwise.} \end{cases} \tag{20}$$

However, such a DP algorithm does not necessarily achieve the optimal solution, since the "weight" of each item is not fixed and even may become negative with different items selected. Consider the whole procedure of the DP algorithm as a tree. During the recursion, once the current combination (e.g., $\mathbf{x} = [1,0,1,0,0,\ldots,0]$) exceeds the threshold, this branch will be cut because all the combinations derived from this branch are regarded to exceed the threshold as well. However, it is possible that there is a combination satisfying the constraints in a subtree due to the items with a negative weight. Although the DP algorithm can cope with the negative weight situation in knapsack problems by preprocessing, it is hard to process "weights" in advance for our problem because of their inter-dependency.

Besides, the DP algorithm has an exponential time complexity. It is applicable when the data is of low scale only. However, a social actor may have hundreds and even thousands of social relations, which make the DP algorithm fail to find out a solution within an acceptable time limit.

To design an efficient algorithm for social network data sharing, both protection performance and computational complexity should be taken into account. In our work, we first propose a heuristic method to obtain a feasible solution to the original combinatorial problem. In order to reduce the high complexity brought by the high-dimensional social relations, we then simplify the original social relation disclosure problem into a d-KP problem.

4.2 Efficiency-based Privacy-Preserving Disclosure

For the knapsack problem and its variants, a great number of heuristic algorithms are proposed to obtain feasible solutions approaching the optimal one with relatively low time complexity [23]. Among them, the greedy algorithm is the most intuitive with a reasonable performance. To obtain a feasible solution with polynomial time complexity, we propose the EPPD algorithm to solve the knapsack-like problem.

We define the total contribution of the selected items T_{sel} to social actor n's secret s as the self privacy disclosure rate:

$$w_s(T_{sel}) = \Phi(u, s, T_{sel}) \tag{21}$$

$$= \frac{|\cap_{t' \in T_{sel}} \mathcal{N}_{t'} \cap \mathcal{N}_s|}{|\cap_{t' \in T_{sel}} \mathcal{N}_{t'}|}. \tag{22}$$

Accordingly, the contribution (weight) of a single item t to social actor u's secret s with selected items T_{sel} can be computed as the increment of the total contribution, i.e.,

$$w_{t,s}(T_{sel}) = \Phi(u, s, T_{sel} \cup \{t\}) - \Phi(u, s, T_{sel}). \tag{23}$$

The main idea of our algorithm is to compare all edges (attribute links or social relations) and find out the most suitable one at each time. If all constraints are satisfied after adding the selected item, then add it to the result list; otherwise, drop it and repeat the former steps. The termination condition is that all the edges are visited, and no more edges can be added into the result set to keep all the constraints satisfied. Intuitively, an edge with a higher value and a lower contribution to each disclosure rate constraint is preferable. Here, we define the *efficiency* of an edge as its value-to-weight ratio. Since there are usually multiple constraints with different thresholds in our problem, it is not reasonable to add up all the "weights" for different secrets directly. Instead, we need to consider the magnitudes and importance of the constraints and compute the efficiency as the following general form:

$$\rho_i = \frac{p_i}{\sum_{j=1}^{m} b_j w_{i,j}}, \tag{24}$$

where b_j is the importance coefficient of the jth constraint and $w_{i,j}$ is the total weight (contribution) with item i to secret j. We use the total weight instead of the weight of each item defined in (23) so as to avoid the negative weight problem. In this way, the efficiency e is always non-negative.

The coefficient b_j can be changed for different purposes to reach a more suitable solution to the given scenario (e.g., a secret with a higher priority can have a smaller b_j). In the experiment shown in Section 5, we only take the magnitudes of constraints into account and let $b_j = 1/\theta_j$. The pseudo-code is shown in Algorithm 1.

From the code, we can see that different from greedy heuristics for knapsack problems which require sorting all items according to a certain criteria at the beginning, the proposed algorithm for the edge masking problem needs to find out the most suitable edge in each iteration, because the "weight" of each item changes after each iteration. Intuitively, in each iteration, we select the attribute or social

Algorithm 1 EPPD Algorithm

Input: Item value list $\vec{p} = \{p_1, p_2, \ldots, p_n\}$, secret neighborhood set list $\vec{S} = \{S_1, S_2, \ldots, S_m\}$, item neighborhood set list $\vec{N} = \{N_1, N_2, \ldots, N_n\}$, and secret threshold list $\theta = \{\theta_1, \theta_2, \ldots, \theta_m\}$

Output: Maximum value $p_{\max}$, result set Sel

```
C ← V_N, Sel ← ∅, V_max ← 0, l⃗ ← [1, 2, ..., n]
while l⃗ ≠ ∅ do
    ρ_max ← −1, s ← −1, w_sel ← ∅
    for each i ∈ l⃗ do
        for j ← 1, 2, ..., m do
            w_j ← |C∩N_i∩S_j| / |C∩N_i|
        end for
        ρ ← p_i / Σ_{k=1}^{m} w_j/θ_j
        if ρ > ρ_max then
            s ← i, ρ_max ← ρ, w_sel ← w
        end if
    end for
    if w_j <= θ_j, ∀j ∈ {1, 2, ..., m} then
        Sel ← Sel ∪ {s}, C ← C ∩ N_s, p_max ← p_max + p_s
    end if
    l⃗.remove(s)
end while
return p_max, Sel
```

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

relation with the highest efficiency, which means that it provides more increment in utility with leaking less information about the secrets. It helps the EPPD algorithm to achieve a high utility while satisfying all the privacy requirements. Moreover, it also naturally supports the case that categorical attributes have different granularities. As long as an attribute a with the social node neighbor set N_a is selected, all the attributes with the social node neighbor set which is the superset of N_a (i.e., the parent attribute in the attribute tree) will be selected because they do not leak more information about the secrets. For example, if the attribute "live in Mountain View, CA" has been selected, the attribute "live in the United States" will be also selected into the result set.

The EPPD algorithm for the social relation masking problem in undirected networks is slightly different. Since the problem is solved as a whole, there may be hundreds of constraints. But actually, the existence of a certain edge only affects two nodes' constraints. With this property, each time we only need to calculate the related constraints and avoid much redundant computation. Therefore, the efficiency of an undirected edge $e = (n_1, n_2)$ can be calculated as follows.

$$\rho_e = \frac{p_e}{\sum_{j=1}^{m_1} b_j w_{n_1,j} + \sum_{j=1}^{m_2} b_j w_{n_2,j}}, \tag{25}$$

where m_1 is the number of social actor n_1's secrets and m_2 is that of social actor n_2.

Computational Complexity: In Algorithm 1, we calculate the efficiency of each item with set intersection operations. Assume that the time complexity of each set intersection operation is $O(|V_N|)$. In the attribute disclosure problem, the time complexity of the EPPD algorithm for **each node actor** is $O(n^2 m \cdot |V_N|)$, where n is the number of attributes and m is the number of secrets. In the practical attribute disclosure problem, the scale of n (the attribute number of a social actor) is usually related to $|V_A|$, which is far less than the scale of $|V_N|$. In the undirected social relation disclosure problem, the time complexity of the EPPD algorithm for **the whole network** is $O(|S| \cdot |V_N| \cdot |E_N|^2)$ where $|S|$ is the total number of all constraints/secrets. When the social network becomes very large, although within polynomial time complexity, it may consume much time to find a feasible solution. This drives us to further simplify the original problem and find an undirected social relation disclosure strategy to obtain a solution with a lower complexity.

4.3 d-KP Simplification

The main factor that causes the high complexity is that the self privacy disclosure considers all the correlations among attributes and social relations. Omitting the correlations and considering all the public information (conditionally) independent, on one hand, may weaken the privacy requirements because two public attributes/social relations may provide more information than the sum of the information provided individually. On the other hand, it will also simplify the problem and fix the weight of each public attribute or social relation. For example, the countermeasures against inference attacks proposed by [16] adopts the Naive Bayes model, which treats all public attributes and social relations conditionally independent, and aims to minimize the classification accuracy. The disclosure strategy is to follow the Naive Bayes model by hiding the attributes and social relations with a higher likelihood. Therefore, in order to further reduce the computation complexity of the undirected social relation disclosure problem, one possible solution is to simplify the original problem with independent assumption.

Since the knapsack problem and its variants have been well-studied for decades, we try to transform the original relation disclosure problem into a multi-dimensional knapsack problem. The key point is how to fix the weight of each item. We can rewrite the constraints (18) and assign a fixed weight to each public attribute. To determine the weight of each relation, we assume that the correlations among relations can be neglected (independent assumption), which is identical to the situation where the adversary only knows the secret distribution in each ego network and the degree of each social actor.

First, we consider the basic form of a single constraint. Given a node u, a secret s and the social relation indicator vector $\mathbf{x}$, let $\vec{v} = \{v_1, v_2, \cdots, v_p\}$ denote the neighbor node of u that whose corresponding social relation's indicator x equals 1, and then we have

$$\Phi_{\mathcal{N}}(u, s, \mathbf{x}) = \Pr\{t_u(s) = 1 | v_1, \ldots, v_p\} \tag{26}$$

$$= \Pr\{t_u(s) = 1\} \prod_{i=1}^{p} \frac{\Pr\{v_i | t_u(s) = 1\}}{\Pr\{v_i\}}, \tag{27}$$

where $\Pr\{v_i\}$ is the probability of connecting to node v_i.

If we substitute (27) into (18) and take the natural logarithm of both sides, we can obtain

$$\sum_{i=1}^{p} \ln \frac{\Pr\{v_i | t_u(s) = 1\}}{\Pr\{v_i\}} + \ln \Pr\{t_u(s) = 1\} \leq \ln \theta. \tag{28}$$

Then, for a node u, the weight of each social relation e is now fixed. If e is the edge connecting to u, the weight is equal to the mutual information between two events, "u is connected to the other node v of e" and "u_k has the secrets of $s_{k,j}$" respectively; otherwise, it is equal to 0. Let $I(e; s_{k,j})$ denote the mutual information between social relation e and node u_k's secret $s_{k,j}$, i.e.,

$$I(e; s_{k,j}) = \begin{cases} \ln \frac{\Pr\{v, t_{u_k}(s_{k,j}) = 1\}}{\Pr\{v\} \Pr\{t_{u_k}(s_{k,j}) = 1\}} & u \in e \\ 0 & u \notin e \end{cases} \tag{29}$$

Note that the "mutual information" defined here is slightly different from that between two random variables. The former can be either non-negative or negative while the latter is always non-negative. A positive "mutual information" implies the contribution to inferring the secret, while a negative one implies the misguidance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

TABLE 2: Secret Settings

Dataset	Attribute name	Category	# of actors
Facebook	School 538	School	631
	Birth year 5	Birthday	372
	Location 84	Hometown	365
	Concentration 14	Concentration	368
Google+	Google Inc.	Institution	1,123
	Manager	Job Title	975
	New York	Location	976

Then we obtain the d-KP simplified version of our proposed optimization problem as follows.

$$\max_{\mathbf{x}} \quad \sum_{i=1}^{|E_N|} p_i x_i \tag{30}$$

$$\text{s.t.} \quad \sum_{i=1}^{|E_N|} I(e_i; s_{k,j}) x_i \le \theta'_{k,j},$$

$$j = 1, \ldots, |\mathcal{S}_{u_k}|, k = 1, \ldots, |V_N|, \tag{31}$$

$$x_i \in \{0, 1\},\ i = 1, \ldots, |E_N|, \tag{32}$$

where $\theta'_{k,j}$ is the new threshold which can be calculated by

$$\theta'_{k,j} = \ln(\exp(\epsilon) + \frac{\delta_{k,j}}{\Pr\{t_{u_k}(s_{k,j}) = 1\}}). \tag{33}$$

The intuition of d-KP transform is to use the information gain as the fixed weight of each relation, which has been widely adopted for feature selection in machine learning. The constraints are to limit the total information gain, which the adversary acquires about the targeted secrets from all disclosed relations, to a small number. Also, we can keep the original privacy constraints and adopt the greedy algorithm for d-KP as the disclosure strategy, selecting the edge $e = (m, n)$ with the minimum weight-price rate (ρ_e^{-1}) which is calculated by

$$\rho_e^{-1} = \frac{\sum_{j=1}^{|\mathcal{S}_m|} I(e; s_{m,j}) + \sum_{j=1}^{|\mathcal{S}_n|} I(e; s_{n,j})}{p_e}. \tag{34}$$

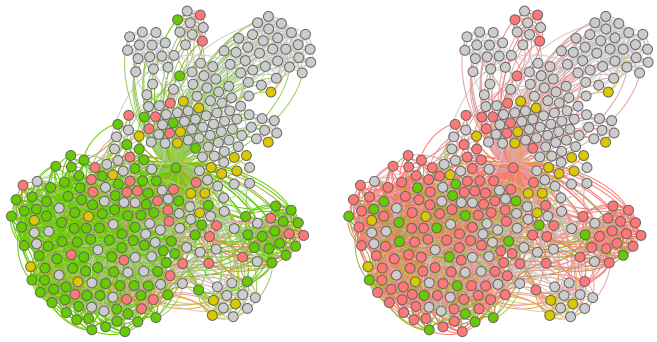
The reason for using ρ_e^{-1} instead of ρ_e is that the information gain can be non-positive in the d-KP simplified problem.

Computational Complexity: The common methods to solve d-KP can be applied to this problem. For our experiments, we use the greedy heuristics to solve the transformed d-KP whose time complexity is $O(|S| \cdot |E_N|)$ [23] with all items sorted in advance. Comparing to the EPPD algorithm, the d-KP disclosure algorithm computes the weights only once and does not need to compute them again during the iterations, which greatly decreases the time complexity.

On one hand, the d-KP disclosure algorithm naturally counters the inference attacks by machine learning. On the other hand, the fixed weight greatly reduces the time complexity with certain protection performance loss compared to directly solving the original relation disclosure problem. Therefore, the d-KP disclosure algorithm is suitable for strict privacy protection which requires a fast response.

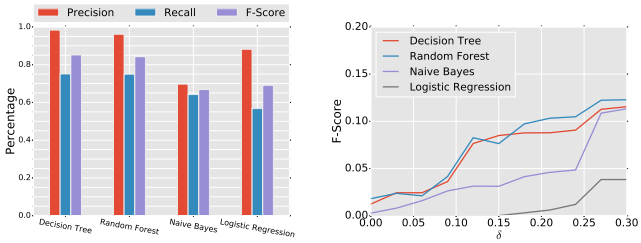
5 Experiment Results

To evaluate the performance of our proposed social network publishing methods, we conduct several experiments on real-world social network datasets with different utility settings and a variety of inference attack algorithms.



(a) Before using EPPD (b) After using EPPD

Fig. 3: Results for inferring the social actors who attended the school 50 in a Facebook Ego Network before and after using the EPPD Algorithm for attribute disclosure. Green: True Positive, Grey: True Negative, Yellow: False Positive, and Red: False Negative.



(a) Performance of local classifiers before using EPPD (b) F-Scores vs δ of local classifiers after using EPPD

Fig. 4: Inference attacks via profile attributes on School 538

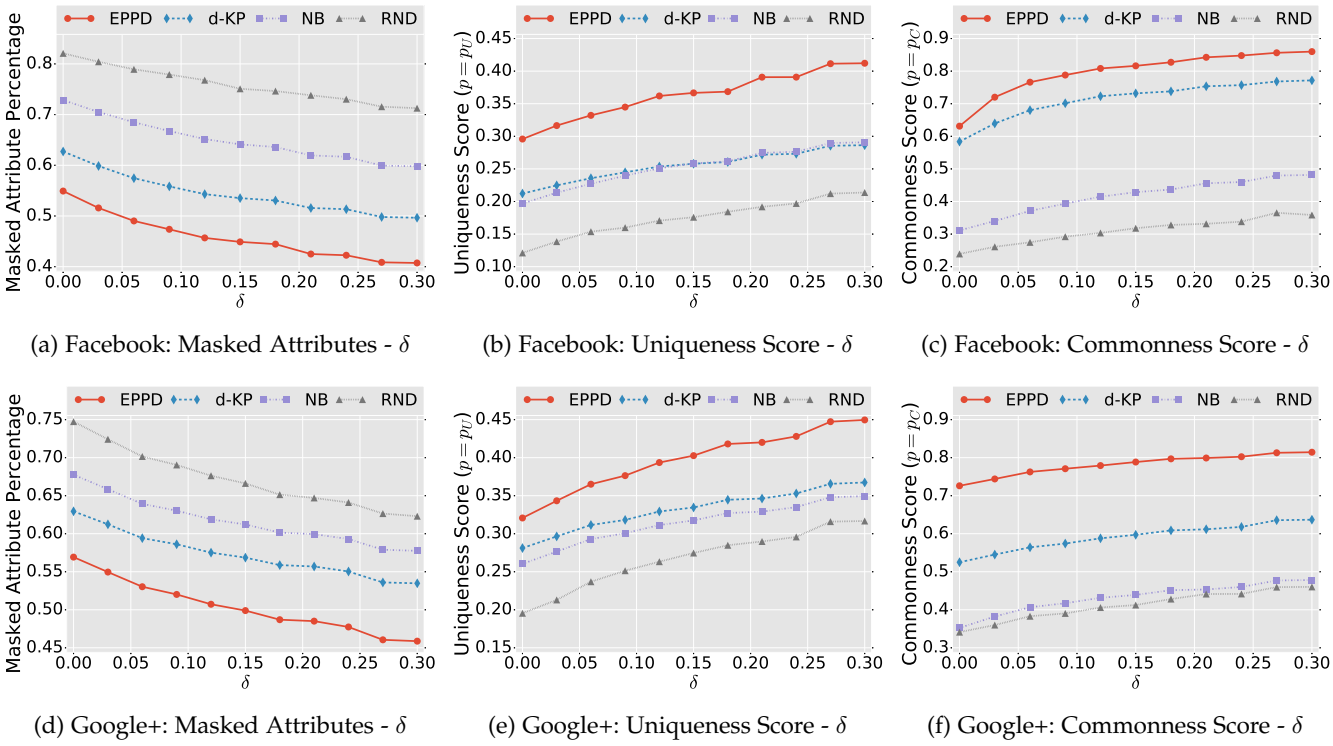
5.1 Methodology

5.1.1 Datasets

We apply our algorithms in 2 real-world social network datasets published by Stanford Network Analysis Project (SNAP) [24]. One is the Facebook dataset, including 4,039 undirected social actors and 88,234 social relations, where user profiles consist of 11 categories (i.e., *gender*, *birthday*, *location*, *hometown*, *work*, *education*, etc.). For some categories, there are several sub-categories (e.g., *employer* and *start date* belong to *work* category). The other is the Google+ dataset, including 17,258 social actors, 1,344,555 directed social relations, where user profiles consist of 5 categories (i.e., *gender*, *institution*, *job title*, *location*, and *university*). Note that attributes in the same category are not necessarily mutually exclusive, as they may coexist in the same user profile (e.g., a user can speak two or more languages).

In the experiment, we select several attributes as the secrets with a moderate sample size (the attributes connected with around 5–10% of social nodes in the datasets) from different categories so that the inference attack can be successfully launched on the original dataset with sufficient positive training data. The detailed secrets settings are shown in the Table 2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing



(a) Facebook: Masked Attributes - δ (b) Facebook: Uniqueness Score - δ (c) Facebook: Commonness Score - δ

(d) Google+: Masked Attributes - δ (e) Google+: Uniqueness Score - δ (f) Google+: Commonness Score - δ

Fig. 5: Profile attribute utility in Facebook and Google+ datasets

5.1.2 Disclosure Methods for Comparison

In order to show the effectiveness of our proposed methods, we introduce some commonly used and benchmark methods for comparison. When we compare the utility metrics, all the algorithms are supposed to use the same privacy constraints introduced in Section 3.1. To determine ϵ, the self privacy disclosure for a binary attribute connected with 10% social actors is supposed not to be larger than 0.5 at $\delta = 0$, where ϵ should be smaller than $\ln(0.5/0.1) \approx 1.6$. To be specific, all ϵ's for different secrets are uniformly set to a moderate value 0.5, and we mainly study the relationship between relaxation variable δ and different metrics. Furthermore, the range of δ is from 0 to 0.3, since 0.3 is already quite high (loose) according to the sample size of the selected secrets.

- *Random Mask (RND)*: Randomly remove attribute or social relation until all of the protection constraints are satisfied. In our experiments, the results of random masking are the average of 100 runs.
- *Naive Bayes Mask (NB)*: Heatherly et al. [16] proposed a masking algorithm based on Naive Bayes classifier to defend against the inference attacks. The basic idea is to remove the attributes and social relations with higher likelihood.
- *d-KP*: In our work, the d-KP algorithm is mainly proposed to solve the undirected social relation disclosure problem. However, it can be also applied to the other disclosure problems with evaluating the information gain of the attributes and social relations as the disclosure criteria.

5.1.3 Algorithms for Inference Attacks

To test the protection effects, we use a variety of classification algorithms to conduct inference attack via either public attributes (local classifier) or social relations (relational classifier). The local classifiers include Decision Tree, Random Forest, Gaussian Naive Bayes and Logistic Regression with the default parameter settings in the Scikit-learn library[1]. The relational classifiers include class-distribution relational neighbor (CDRN), weighted-vote relational neighbor (WVRN) and network-only link-based classification (NOLB) implemented in Netkit-SRL library[2], which were proposed by S. A. Macskassy et al. [25]

To launch the inference attack, we have the following experiment settings. For local classifiers, the adversary is assumed to have the whole dataset as its training set, and then conduct the inference attack on the published dataset with the trained model. For relational classifiers, the adversary is assumed to have the half of the dataset and published network as its training set, and then conduct the inference attack on the remaining part of the dataset with the trained model. The different experiment settings are due to the different design principles. The relational classifiers rely on the identity of social actors while the local ones do not. It makes no sense for the relational classifier to train on the whole dataset with all actors' secrets already known.

1. http://scikit-learn.org/stable/
2. http://netkit-srl.sourceforge.net/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

5.2 Attribute Disclosure Results

5.2.1 Protection Performance

The protection performance of our proposed disclosure algorithm depends on the self privacy disclosure constraint, which is defined in Section 3. As long as the self-privacy disclosure rate is smaller than the constraint threshold, we regard the published social network as a privacy-preserving one. In this part, we mainly study the protection performance of our EPPD algorithm under different δ, because the self privacy disclosure rate of applying EPPD is closer to the constraint threshold compared to the other algorithms.

We first show an example of the inference attack based on the Decision Tree classifier on a small Facebook ego network with 348 users in Fig. 3, where the targeted secret is the education attribute School 50 (with 154 users having this attribute). Before we apply the EPPD algorithm, the adversary successfully infers 119 of 154 users with the secret (Precision: 83.80%, Recall: 77.27% and F-Score: 80.41%). After applying the EPPD algorithm ($\delta = 0.3$), the adversary fails to find out most positive instances with only 23 inferred correctly (Precision: 14.94%, Recall: 16.20% and F-Score: 15.54%). The performance of the inference attack decreases significantly, which means that the EPPD algorithm is able to defend against the inference attack effectively. For the experiment on the large dataset, we mainly study the F-Score of the classification results because it takes both precision and recall into account.

Fig. 4 shows the experiment on inferring the secret School 538 in the Facebook dataset. As shown in Fig. 4a, the original F-Score of 4 classifiers are 85.17% (Decision Tree), 84.24% (Random Forest), 66.83% (Naive Bayes), and 69.05% (Logistic Regression), respectively, where the random guess is around 15.62%. However, after applying the EPPD algorithm, the F-Scores of these four local classifiers are no larger than 15% even with a very loose $\delta = 0.3$. With critical attributes concealed, the published user information will mislead the local classifier to make an inaccurate prediction. Therefore, the local classifiers can hardly work on the processed dataset. When the privacy constraints are strict $\delta = 0$, almost every social actor with the targeted secret is well protected with all F-Scores around 1% only. All these results show that the EPPD algorithm can well defend against the inference attack via a variety of local classifiers.

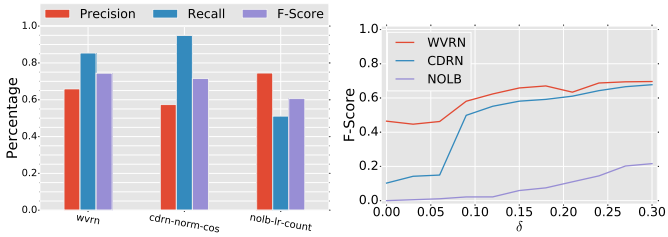
5.2.2 Utility Comparison

To evaluate the performance of attribute disclosure, we mainly compare the utility scores and the percentage of masked attributes under different social network data sharing algorithms. In our experiment, we adopt the normalized value of each utility score p (e.g., uniqueness score and commonness score) which is calculated as follows.

$$\mathcal{U} = \frac{\sum_{u \in V_N^*} \sum_{i=1}^{|\mathcal{P}_u|} p_i x_i}{\sum_{u \in V_N^*} \sum_{i=1}^{|\mathcal{P}_u|} p_i}, \tag{35}$$

where $V_N^* = \{u | u \in V_N, \mathcal{S}_u \neq \emptyset\}$ is the set of the affected social actors having privacy concerns. For an algorithm, a higher utility score indicates that more valuable non-sensitive attributes are shared.

Fig. 5a and 5d show the attribute disclosure results of experiments on the Facebook and Google+ datasets. For all



(a) Performance of relational classifiers before using EPPD (b) F-Scores vs δ of relational classifiers after using EPPD

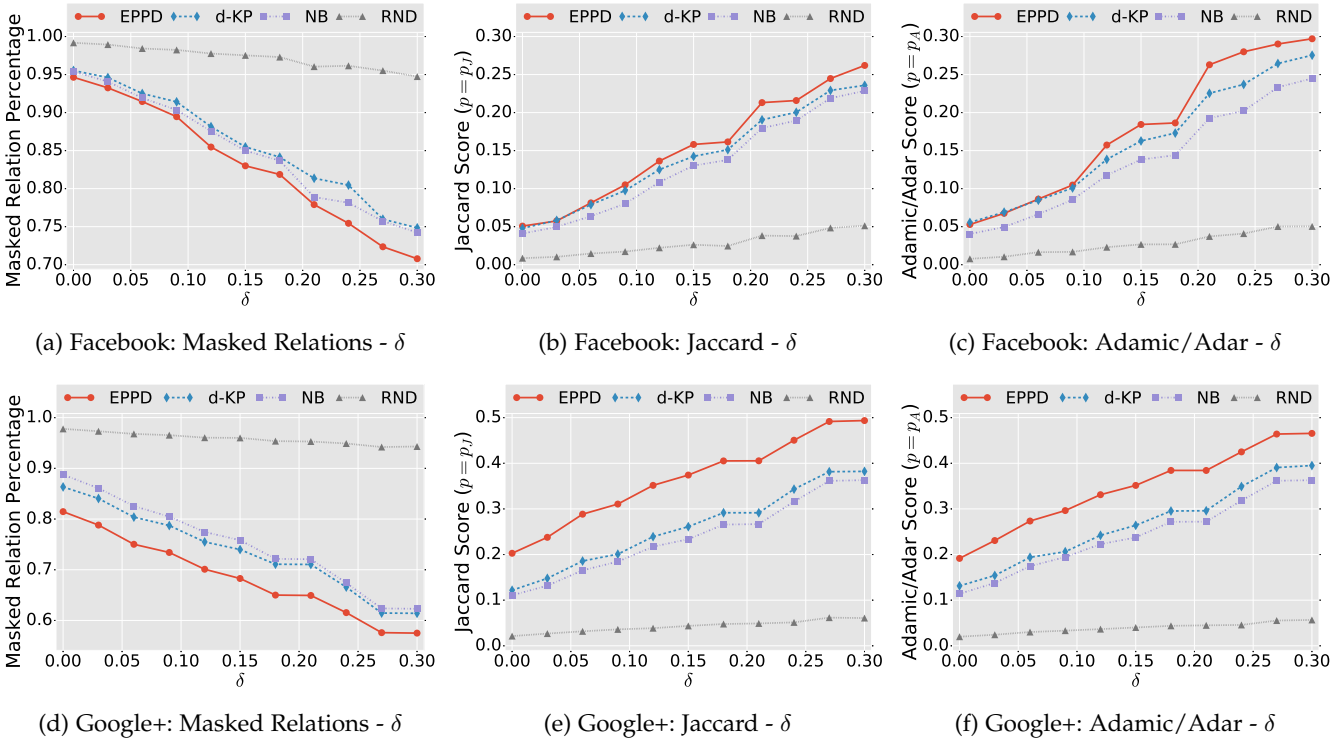
Fig. 6: Inference attacks via social relations on School 538

the four algorithms, the percentage of the masked attributes decreases with the increment of δ. According to the results in the Facebook dataset, the EPPD algorithm always has the best performance, where only 40.74% of the public attributes are masked at $\delta = 0.3$, while the other three algorithms need to mask 49.66% (d-KP), 59.79% (NB) and 71.27% (RND) of the public attributes. Even under the strict protection constraints at $\delta = 0$, the EPPD algorithm only masks 55% of the public attributes. Besides, the d-KP algorithm is also better than the Naive Bayes Masking algorithm with fewer attributes masked. Similar results are also observed from the experiment on the Google+ dataset. Combining with results of protection performance experiments, the EPPD algorithm can realize a good protection effect by removing as few critical attributes as possible.

For the utility score experiments, we try the two different utility settings, namely the uniqueness score and the commonness score introduced in Section 2.5. Generally, the utility scores always increase with the increment of δ since a loose privacy constraint allows more information published. Fig. 5b and 5e compare the uniqueness scores of 4 algorithms in the Facebook and Google+ datasets. Intuitively, an attribute with a higher uniqueness score may imply more information about whether a social actor has a certain secret, which conflicts the idea of making a social actor's profile indistinguishable. Therefore, all the four algorithms have low uniqueness scores. Besides, the d-KP, an algorithm based on Information Gain, has a close performance to the Naive Bayes Masking in the Facebook dataset while it has a slightly higher score in the Google+ dataset. However, the EPPD algorithm has the best performance among the four algorithms with around 40-50% performance increase over the Naive Bayes Masking.

In contrast with the uniqueness score, the commonness score weighs more on the common attributes. As shown in Fig. 5c and 5f, both EPPD and d-KP algorithms have significantly higher scores than Naive Bayes Masking. In the Google+ dataset, the EPPD algorithm performs much better than the other three algorithms at any δ while the Naive Bayes Masking algorithm is just slightly better than Random Masking since it does not set the utility score as its objective. In summary, for the attribute disclosure problem, the EPPD algorithm is more desirable since it has a higher utility score and masks fewer attributes with an acceptable time complexity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing



(a) Facebook: Masked Relations - δ (b) Facebook: Jaccard - δ (c) Facebook: Adamic/Adar - δ

(d) Google+: Masked Relations - δ (e) Google+: Jaccard - δ (f) Google+: Adamic/Adar - δ

Fig. 7: Social relation utility in Facebook and Google+ datasets

5.3 Social Relation Disclosure Results

5.3.1 Protection Performance

Similar to the attribute disclosure experiments, we first study the protection performance of the EPPD algorithm against three different relational classifiers, WVRN, CDRN, and NOLB respectively. Although they all focus on the ego network of the targeted node, they have different classification principles which lead to different performances, where WVRN first assigns weights to the neighboring nodes then to get the weighted average probability of owning the secret, CDRN is based on the secret distribution in the ego network and NOLB uses logistic regression with labels of the neighboring nodes as feature vectors.

Fig. 6a shows the inference results of the three relational classifiers on School 538 before applying the EPPD algorithm. From the figure, we can see that WVRN has a slightly better performance with a higher F-Score 74.41%, compared with CDRN (71.53%) and NOLB (65.93%). CDRN features a high recall score 94.92% while NOLB has a relatively high precision score 74.54%. Fig. 6b shows the change of F-Scores of the three relational classifiers with the increment of δ. The performance of NOLB decreases significantly at any δ. However, unlike the local classifiers, the F-Scores of WVRN and CDRN do not decrease dramatically at a loose $\delta = 0.3$ (WVRN: 69.61% and CDRN: 67.75%), and they are still even better than that of NOLB attacking the original dataset. When $\delta \leq 0.06$, the F-Scores of the two classifiers are smaller than 0.5, making them ineffective in the inference attack.

The protection experiment results imply that a loose privacy constraint for social relation disclosure is still likely to expose the users' secrets to the threat of inference attacks by means of relational classifiers. Similar results were also observed in [16] where the removal of a number of social relations does not lead to a significant performance decrease for the relational classifiers. Thus, it is recommended to set a small value for the privacy threshold (both δ and ϵ) to protect against social relation based inference attacks.

5.3.2 Utility Comparison

As discussed in Section 3, the directed and undirected social relation disclosure problems are quite different from each other. Here, we conduct both experiments with the Facebook dataset (undirected) and the Google dataset (directed). Besides the percentage of masked relations, we also use Jaccard coefficient and Adamic/Adar score introduced in Section 2.5 for performance evaluation. We also calculate the normalized value of each utility score p as follows.

$$\mathcal{U} = \frac{\sum_{i=1}^{|E_N^*|} p_i x_i}{\sum_{i=1}^{|E_N^*|} p_i}, \tag{36}$$

where $E_N^* = \{e | e = (u, v) \in E_N, \mathcal{S}_u \neq \emptyset \text{ or } \mathcal{S}_v \neq \emptyset\}$ is the set of the affected edges with the social actors having privacy concerns.

Fig. 7a and 7d show the masked relation percentage the processed networks under different disclosure algorithms. Compared with the attribute disclosure, the social relation disclosure needs to mask much more edges, especially for the undirected one. As shown in the figure, when $\delta = 0$, nearly 95% affected social relations in the Facebook network need to be removed to satisfy the privacy guarantee while in the Google+ network, the EPPD algorithm can only retain about 18.54% affected social relations (d-KP: 13.67% and NB: 11.25%). Although the EPPD algorithm can keep around 30% social relations at a loose $\delta = 0.3$, the performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

of the inference attack does not decrease significantly. That is to say, to protect the privacy effectively, masking a great number of relations is essential, which leads to a limited operation space for privacy-preserving disclosure algorithms. For this reason, the performances of the three disclosure algorithms look similar. Nevertheless, the EPPD algorithm still has a slightly better performance over d-KP and NB.

Both Jaccard coefficient and Adamic/Adar score assign a weight to each social relation according to its importance based on the similarity between the two connected nodes. However, the former describes the similarity by the common friends (structure similarity) while the latter emphasizes the similarity between the two nodes' profiles (profile similarity). Fig. 7b and 7e illustrate the Jaccard results in both directed and undirected social networks. In the undirected Facebook network, the performances of EPPD and d-KP are very close and still slightly better than NB at a strict $\delta < 0.06$. And in the directed Google+ network, the EPPD algorithm has higher utility scores compared with the other two algorithms. Similar results are also observed in the Adamic/Adar experiments as shown in Fig. 7c and 7f. In conclusion, the utility performance of the EPPD algorithm is always good since it is designed to keep the important relations as many as possible. However, in the undirected social relation disclosure experiments, the d-KP algorithm, which simplifies the original disclosure problem and reduces the computational complexity, has a very similar performance with the EPPD algorithm at a strict δ. In this case, we can use the d-KP algorithm to replace the EPPD algorithm for a faster response and fairly good results.

5.4 Summary

From the experiment results, we can conclude that the EPPD algorithm can effectively defend against different kinds of inference attacks via public attributes and social relations. Under the same security requirements, the EPPD algorithm can achieve a higher utility score than the state-of-the-art algorithm for both attribute disclosure and social relation disclosure problems. As shown in the undirected social relation experiments on Facebook, the d-KP algorithm has a similar performance with the EPPD algorithm with a lower complexity. Thus, we can also consider the d-KP algorithm as a substitute of EPPD in the undirected social relation disclosure problem.

6 Related Work

Privacy-preserving Social Network Publishing: Different from the work focusing on the privacy-preserving distributed data aggregation [22], [26], publishing a social network usually takes the entire network into consideration directly. There are extensive works on privacy-preserving social network data sharing and publishing, which mainly focused on anonymization techniques. Li et al. [14] proposed a graph-based privacy-preserving data publishing framework to construct several subgraphs and perform the existing anonymity operations independently for each subgraph. Wang et al. [27] studied how to outsource social networks with indistinguishability. The introduction of differential privacy [21] also provides a solid theoretical foundation for social network data publishing. Jorgensen et al. [28] involved differential privacy guarantees into attributed social graph publishing, and Day et al. [29] proposed a differential privacy based graph degree distribution publishing method.

Controlled Information Sharing in OSNs: One of the typical security and privacy issues in OSNs is how to protect data from unauthorized access [30]. To address this issue, there are extensive works on the access control models, which concentrate on how to share the social network data according to identity, relationship, and data sensitivity [31], [32], [33]. They mainly studied the design of access control policies to secure the private social network information based on the trust among users. In addition, for the specific resources like photos, Xu et al. [34] proposed a distributed consensus-based method to control the photo sharing with friends by using an efficient facial recognition system. However, few of these works take the inference attack into account, which can be conducted via authorized access. Our work is from the novel perspective of the correlations between public and private information, which makes it possible to integrate with the existing access control models to defend against the inference attack.

Inference Attack: Inference attack has been widely studied to infer user demographics [5], [35], social roles [7], [36], hidden attributes [4], [6], [37], user activities [38], etc., in online social networks. Most of the inference attacks extract the features from the published data as the input of trained models to obtain the most probable secrets. To describe attackers' capabilities, Qian et al. [13], [15] used the knowledge graph where each relation connecting two entities is a piece of knowledge. In our work, the background knowledge of an attacker is captured by a social-attribute network [11], [12] where each piece of knowledge is a tree-like inference path involving at least 3 nodes. To defend against the inference attack, Heatherly et al. [16] studied the inference attack based on Naive Bayes and introduced a correspondent protection approach. Different from our work, the principle of its masking algorithm is to remove the most highly indicative attributes and social relations without considering the correlation among public information, which makes it hard to satisfy a variety of utility metrics and the inference attacks based on other statistical learning methods. For our EPPD algorithm, we consider the correlations among public attributes or social relations, and consider customized utility values of them.

7 Conclusions

This paper investigated the online social network data sharing with defense against the inference attack and formulated the optimization problem which maximizes the utility with privacy guarantee and user privacy concerns. In the paper, we have proposed two different methods to address the problem. One is the EPPD algorithm which is based on the greedy heuristics to directly obtain a feasible and good solution to the original problem, while the other is the d-KP approximation algorithm which transforms and simplifies the undirected social relation disclosure problem into a d-KP problem. Extensive experiments on real-world datasets demonstrated that our proposed methods are efficient and

effective to defend against the inference attack, and especially the EPPD algorithm substantially outperforms the existing masking techniques while the d-KP approximation algorithm has a lower computational complexity, and it is applicable when dealing with the undirected social relation disclosure problem.

For the future work, we will further study the utility metrics to consider more complicated situations, such as non-linear utility objectives and attributes with different granularities. For the social relation disclosure problem, we will also investigate the correlations among structural or community information and user privacy, and makes the corresponding protection algorithms to defend against the inference attack based on these aspects.

ACKNOWLEDGMENT

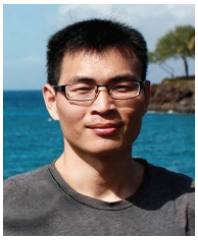
The authors would like to thank the editor and the anonymous reviewers for their helpful and constructive comments, which have helped us improve the manuscript significantly. This work was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC), Canada Foundation for Innovation (CFI), BC Knowledge Development Fund (BCKDF), and the Natural Science Foundation of China (NSFC) under grant 61773257, 61761136012.

REFERENCES

[1] A. M. Kaplan and M. Haenlein, "Users of the world, unite! the challenges and opportunities of social media," *Business Horizons*, vol. 53, no. 1, pp. 59–68, 2010.
[2] C. Dwyer, S. Hiltz, and K. Passerini, "Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace," *AMCIS 2007 proceedings*, p. 339, 2007.
[3] D. Hardt, "The OAuth 2.0 authorization framework," 2012.
[4] Q. Fang, J. Sang, C. Xu, and M. S. Hossain, "Relational user attribute inference in social media," *IEEE Transactions on Multimedia*, vol. 17, no. 7, pp. 1031–1044, 2015.
[5] Y. Dong, Y. Yang, J. Tang, Y. Yang, and N. V. Chawla, "Inferring user demographics and social strategies in mobile social networks," in *Proc. of ACM SIGKDD*, 2014.
[6] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: Inferring user profiles in online social networks," in *Proc. of ACM WSDM*, 2010.
[7] Y. Zhao, G. Wang, P. S. Yu, S. Liu, and S. Zhang, "Inferring social roles and statuses in social networks," in *Proc. of ACM SIGKDD*, 2013.
[8] X. Wu, X. Ying, K. Liu, and L. Chen, "A survey of privacy-preservation of graphs and social networks," in *Managing and Mining Graph Data*. Springer, 2010, pp. 421–453.
[9] A. L. Young and A. Quan-Haase, "Information revelation and Internet privacy concerns on social network sites: a case study of Facebook," in *Proceedings of the fourth international conference on Communities and technologies*. ACM, 2009, pp. 265–274.
[10] Z. Yin, M. Gupta, T. Weninger, and J. Han, "LINKREC: A unified framework for link recommendation with user attributes and graph structure," in *Proc. of ACM WWW*, 2010.
[11] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar, and D. Song, "Evolution of social-attribute networks: Measurements, modeling, and implications using Google+," in *Proc. of ACM IMC*, 2012.
[12] N. Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. R. Shi, and D. Song, "Joint link prediction and attribute inference using a social-attribute network," *ACM TIST*, vol. 5, no. 2, p. 27, 2014.
[13] J. Qian, X.-Y. Li, C. Zhang, and L. Chen, "De-anonymizing social networks and inferring private attributes using knowledge graphs," in *Proc. of IEEE INFOCOM*, 2016.
[14] X.-Y. Li, C. Zhang, T. Jung, J. Qian, and L. Chen, "Graph-based privacy-preserving data publication," in *Proc. of IEEE INFOCOM*, 2016.
[15] J. Qian, X. Y. Li, C. Zhang, L. Chen, T. Jung, and J. Han, "Social network de-anonymization and privacy inference with knowledge graph model," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2017.
[16] R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "Preventing private information inference attacks on social networks," *IEEE TKDE*, vol. 25, no. 8, pp. 1849–1862, 2013.
[17] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.
[18] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern, "Worst-case background knowledge for privacy-preserving data publishing," in *Proc. of IEEE ICDE*, 2007.
[19] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM TKDD*, vol. 1, no. 1, p. 3, 2007.
[20] J. Liu, L. Xiong, and J. Luo, "Semantic security: Privacy definitions revisited." *Transactions on Data Privacy*, vol. 6, no. 3, pp. 185–198, 2013.
[21] C. Dwork, "Differential privacy: A survey of results," in *Proc. of TAMC*. Springer, 2008.
[22] J. He, L. Cai, and X. Guan, "Preserving data-privacy with added noises: Optimal estimation and privacy analysis," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5677–5690, Aug 2018.
[23] H. Kellerer, U. Pferschy, and D. Pisinger, *Introduction to NP-Completeness of knapsack problems*. Springer, 2004.
[24] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.
[25] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," *Journal of Machine Learning Research*, vol. 8, no. May, pp. 935–983, 2007.
[26] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, "Distributed privacy-preserving data aggregation against dishonest nodes in network systems," *IEEE Internet of Things Journal*, pp. 1–1, 2018.
[27] G. Wang, Q. Liu, F. Li, S. Yang, and J. Wu, "Outsourcing privacy-preserving social networks to a cloud," in *Proc. of IEEE INFOCOM*, 2013.
[28] Z. Jorgensen, T. Yu, and G. Cormode, "Publishing attributed social graphs with formal privacy guarantees," in *Proc. of ACM SIGMOD*, 2016.
[29] W.-Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proc. of ACM SIGMOD*, 2016.
[30] B. Carminati, E. Ferrari, and M. Viviani, "Security and trust in online social networks," *Synthesis Lectures on Information Security, Privacy, & Trust*, vol. 4, no. 3, pp. 1–120, 2013.
[31] B. Carminati, E. Ferrari, and A. Perego, "Enforcing access control in web-based social networks," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 1, p. 6, 2009.
[32] H. Hu, G.-J. Ahn, and J. Jorgensen, "Multiparty access control for online social networks: model and mechanisms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1614–1627, 2013.
[33] Y. Cheng, J. Park, and R. Sandhu, "An access control model for online social networks using user-to-user relationships," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 4, pp. 424–436, 2016.
[34] K. Xu, Y. Guo, L. Guo, Y. Fang, and X. Li, "My privacy my decision: Control of photo sharing on online social networks," *IEEE Transactions on Dependable and Secure Computing*, 2015.
[35] H. Li, H. Zhu, S. Du, X. Liang, and X. Shen, "Privacy leakage of location sharing in mobile social networks: Attacks and defense," *IEEE Transactions on Dependable and Secure Computing*, 2016.
[36] J. Chen, J. He, L. Cai, and J. Pan, "Profiling online social network users via relationships and network characteristics," in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–6.
[37] 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 *Proc. of ACM WWW*, 2009.
[38] J. Song, S. Lee, and J. Kim, "Inference attack on browsing history of Twitter users using public click analytics and Twitter metadata," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 3, pp. 340–354, 2016.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861403, IEEE Transactions on Dependable and Secure Computing

Jiayi Chen is currently a Ph.D. student in the at University of Waterloo, Waterloo, ON, Canada. He received the M.A.Sc degree in the Computer Engineering from University of Victoria, Victoria, BC, Canada in 2017, and the Bachelor of Engineering in Computer Science from Shanghai Jiao Tong University, Shanghai, China in 2015. His research interests involve mobile security and social network privacy.



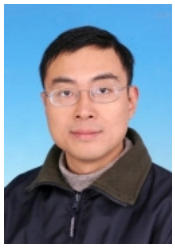
Jianping He (M'15) is currently an associate professor in the Department of Automation at Shanghai Jiao Tong University, Shanghai, China. He received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2013, and had been a research fellow in the Department of Electrical and Computer Engineering at the University of Victoria, Canada, from Dec. 2013 to Mar. 2017.

His research interests mainly include the smart sensing and control, security and privacy theory and its applications, distributed learning and big data.

He serves as an associate editor for the KSII Transactions on Internet and Information Systems. He was also a guest editor for the IEEE Transactions on Automatic Control, International,Journal of Robust and Nonlinear Control and Neurocomputing. He was the winner of Outstanding Thesis Award, Chinese Association of Automation, 2015. He received the best paper award of IEEE WCSP'17 and the finalist best student paper award of IEEE ICCA'17. He is the recipient of China National Recruitment Program of 1000 Talented Young Scholars.

Lin Cai Lin Cai (S'00-M'06-SM'10) received her M.A.Sc. and PhD degrees in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2002 and 2005, respectively. Since 2005, she has been with the Department of Electrical & Computer Engineering at the University of Victoria, and she is currently a Professor. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and Internet of Things.

She was a recipient of the NSERC Discovery Accelerator Supplement (DAS) Grants in 2010 and 2015, respectively, and the Best Paper Awards of IEEE ICC 2008 and IEEE WCNC 2011. She has founded and chaired IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She has served as a member of the Steering Committee of the IEEE Transactions on Big Data, an Associate Editor of the IEEE Internet of Things Journal, IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, EURASIP Journal on Wireless Communications and Networking, International Journal of Sensor Networks, and Journal of Communications and Networks (JCN), and as the Distinguished Lecturer of the IEEE VTS Society. She has served as a TPC symposium co-chair for IEEE Globecom'10 and Globecom'13. She is a registered professional engineer of British Columbia, Canada.



Jianping Pan (S'96-M'98-SM'08) is currently a professor of computer science at the University of Victoria, Victoria, British Columbia, Canada. He received his Bachelor's and PhD degrees in computer science from Southeast University, Nanjing, Jiangsu, China, and he did his postdoctoral research at the University of Waterloo, Waterloo, Ontario, Canada. He also worked at Fujitsu Labs and NTT Labs. His area of specialization is computer networks and distributed systems, and his current research interests include protocols for advanced networking, performance analysis of networked systems, and applied network security. He received the IEICE Best Paper Award in 2009, the Telecommunications Advancement Foundation's Telesys Award in 2010, the WCSP 2011 Best Paper Award, the IEEE Globecom 2011 Best Paper Award, the JSPS Invitation Fellowship in 2012, the IEEE ICC 2013 Best Paper Award, and the NSERC DAS Award in 2016, and has been serving on the technical program committees of major computer communications and networking conferences including IEEE INFOCOM, ICC, Globecom, WCNC and CCNC. He was the Ad Hoc and Sensor Networking Symposium Co-Chair of IEEE Globecom 2012 and an Associate Editor of IEEE Transactions on Vehicular Technology. He is a senior member of the ACM and a senior member of the IEEE.