Privacy Inference on Knowledge Graphs: Hardness and Approximation

Jianwei Qian*, Shaojie Tang†, Huiqi Liu‡, Taeho Jung*, Xiang-Yang Li‡
* Department of Computer Science, Illinois Institute of Technology
† Department of Information Systems, University of Texas at Dallas
‡ School of Computer Science and Technology, University of Science and Technology of China

***Abstract*—The rapid information propagation facilitates our work and life without precedent in history, but it has tremendously exaggerated the risk and consequences of privacy invasion. Today's attackers are becoming more and more powerful in gathering personal information from many sources and mining these data to further uncover users' privacy. A great number of previous works have shown that, with adequate background knowledge, attackers are even able to infer sensitive information that is not revealed to anyone malicious before. In this paper, we model the attacker's knowledge using a knowledge graph and formally define the privacy inference problem. We show its #P-hardness and design an approximation algorithm to perform privacy inference in an iterative fashion, which also reflects real-life network evolution. The simulations on two data sets demonstrate the feasibility and efficacy of privacy inference using knowledge graphs.**

I. Introduction

Today's all-pervasive technological innovations and rapid information propagation have been unprecedentedly bettering our life in all aspects. However, people's personal data have been collected, analyzed, shared and released, which has raised much concern and resistance from the public [9]. Privacy as one of the most primary needs of human beings, guarantees our freedom of expression and lifestyles and the ease of worries about being watched, judged, or harmed. Nevertheless, privacy breach is real and serious, such as the successful de-anonymization of the released AOL search logs in 2006 [1] and the Netflix Prize data set in 2007 [20], and the leak of Ctrip user payment logs in 2014 [34]. These famous leak incidents are the tip of the iceberg; considerable privacy invasions are barely known to the public [2], for example the ongoing customer information selling [9].

This paper concentrates on *privacy inference*, a main type of privacy invasion. It refers to the process of inferring new information about users given that the attacker already possesses some background knowledge. The background knowledge can be gathered from multiple sources such as common sense, real life observations, demographic statistics, and shared or publicly released data sets. We take the gigantic and fast-growing technology companies Facebook and Google as two examples. Facebook's friend recommendation system predicts potential connections based on the "proximity" of users such as geographic location, common hobbies and common neighbors [4]. Google provides us with smart and convenient services but meanwhile large quantities of personal data have been recorded, including locations and mobility traces, calendar schedules, emails, and cloud documents. The rich personal information has been analyzed for inferring users' demographic categories, interest and hobbies, and even itineraries to come, some of which might be thought as private by the target individuals. Both Facebook and Google were once charged by Federal Trade Commission due to privacy issues [9].

The **goal** of this paper is to model and formulate the attacker's prior knowledge and the process of privacy inference attack from a general view. Hopefully, it would reveal the essence of privacy inference to the public and enlighten data protectors to take the appropriate measures for privacy preservation. To this end, we need to address three **challenges**. *First*, privacy itself is comprehensive and difficult to define and privacy leakage is hard to quantify. Various personal information can be private, including many attributes (such as disease, marital status and annual salary) and interconnections between users (*e.g.* affairs and underground organizations). *Second*, it is challenging to apply one single model to all the data forms since the data being attacked can be heterogeneous and arbitrary, such as relational data [18], social network data [21], spatio-temporal data [31], genome data [22] and so on. *Third*, the privacy inference process is hard to model because there are a large diversity of techniques for it, *e.g.* homogeneity attack [18], minimality attack [35], and structure-based de-anonymization attacks [21].

Privacy has always been a heated research topic in the literature. Some works demonstrate a variety of attack techniques for relational data in the data publishing scenario [6], [17] or the interactive statistical query scenario [7], [8]. Others propose plenty of powerful de-anonymization and privacy learning methods on social network data [11], [21]. Our work **differs** from much of the related work in that it abstracts the process of privacy inference in a general fashion.

In this paper, we model the attacker's knowledge using a knowledge graph (in Section II-A) and give four base cases of privacy inference on this model (in Section II-B). We also formally define the privacy inference problem (in Section III-A) and prove its #P-hardness (in Section III-B).

Generally, our contributions can be summarized as follows.

- We present a detailed analysis of the nature of prior knowledge and privacy inference and model them with the knowledge graph model (in Section II).

- We formulate the privacy inference process and prove that it is a #P-hard problem (in Section III), and propose a heuristic method to estimate the inference outcome (in Section IV).
- We implement our privacy inference algorithm and conduct simulations on real world data sets which verifies the effectiveness of the algorithm (in Section V).

II. Preliminaries

A. Background Knowledge & Knowledge Graph

In reality, privacy inference attack is usually performed by an attacker who has possession of some background knowledge [8], [18]. It can be gathered from multiple sources including common sense, statistical data, personal data, and released data sets. The rich and sophisticated data gathered by the attacker are merged and stored in a knowledge graph.

A *knowledge graph* is a heterogeneous graph of all kinds of entities and their relations related to a specific domain or topic [12]. The currently most famous knowledge graphs include Freebase, Google's Knowledge Graph, and Facebook's Entities Graph. In this paper, we model knowledge graph as a directed graph in which each node represents an entity, each directed edge represents a relation between two entities. Moreover, each edge has a relation type indicating the specific relationship, and a probability (referred to as confidence score) indicating the attacker's confidence on the relation. Therefore, each edge in the knowledge graph represents a piece of knowledge of the attacker. The confidence score is either 1 or 0 for exact knowledge, and it is 0.5 for unknown edges. Since all the unknown edges are also added to the knowledge graph, it is always a complete graph. Hereinafter, we denote a knowledge graph as $G = (V, E, P)$ where $V = U \cup A$, U is the user node set, A is the attribute node set, E is the set of relations between nodes, and P is the set of confidence scores on the relations in E. We denote an edge as a quadruplet $e = (s, r, o, p)$, where $s, o \in V$, $r \in E$, $p \in P$, meaning that the attacker believes there is a relation r connecting from the subject s to the object o with the probability of p. The confidence score/probability of an edge e is denoted as $p(e)$. Multiple edges are allowed considering that two entities could be connected through multiple ways. For example, two persons are both coworkers and friends, and a person's location and hometown are the same place.

There are three types of edges/relations in the knowledge graph. *User-to-user (U-U) relations* represent relations between users. They could be uni-directed (star-fan, father-son, employer-employee) or bi-directed (friends, colleagues, classmates, couples). We denoted a U-U relation as $e_{u_1 u_2}$, $u_1, u_2 \in U$. *Attribute-to-attribute (A-A) relations* model the correlations between attributes. They are directed indicating that one attribute is dependent on another. Bayesian network can also be used to model the dependence of attributes. We denoted an A-A relation as $e_{a_1 a_2}$, $a_1, a_2 \in A$. *User-to-attribute (U-A) relations* stand for users' attributes, *e.g.* $(\text{Bob, has occupation of, technician}, 0.9)$. They can be treated as undirected since there is no ambiguity. We denoted a U-A relation as e_{ua}, $u \in U$, $a \in A$. Thus, we have $E = E^{UU} \cup E^{UA} \cup E^{AA}$. There are mass of research works on knowledge graph construction [23], [24] and refinement [25], [30]. Thus in this paper, we assume that the attacker has already constructed a knowledge graph using her prior knowledge.

B. Base Cases of Privacy Inference

Privacy inference (or more accurately *knowledge inference*) refers to the process of inferring new relations between a user and an attribute or another user in the knowledge graph and calculating the probabilities of them being true. There are four basic types of privacy inference as follows.

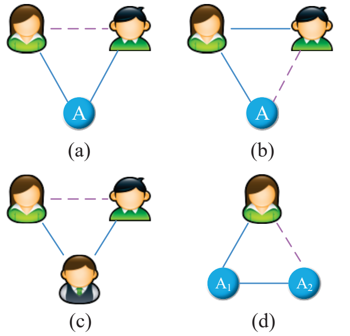


Fig. 1: **Four base cases of privacy inference (triangle inference):** 'A' stands for an attribute. Each triangle inference consists of two known edges (blue arrows) and one unknown edge (dotted purple arrow) to be inferred. The inference is based on common neighbors (attributes or users).

Case a [U-A + U-A → U-U], which means inferring a U-U relation with two U-A relations (similarly hereinafter). As shown in Fig. 1(a), the attacker can infer the relationship between two persons according to their common attributes. For example, working at the same company implies that they are colleagues, and running for the US president implies that they are opponents. Sometimes the inference is not absolutely correct, *e.g.* people sharing the same home address are not necessarily families. There is a probability in the correctness of the inference, referred to as *inference probability*.

Case b [U-U + U-A → U-A]: Fig. 1(b) shows that the attacker can infer whether a user possesses a specific attribute given that he is related to another person who has this attribute. As a simple example, we can infer that Tom also plays for a baseball team given that his teammate Bob plays for that team.

Case c [U-U + U-U → U-U]: As depicted in Fig. 1(c), the attacker can infer the relation between two persons who are both connected to a third person. For instance, two parents of the same kid are very likely to be spouses. Another example is that two persons sharing the same friend could also be friends.

Case d [U-A + A-A → U-A]: Fig. 1(d) indicates that the attacker can infer an attribute of a person based on its dependency on another attribute, *e.g.* education level has a major influence on the amount of salary a person earns.

We need to combine multiple base cases together if there are multiple common neighbors. Specifically, multiple cases of

Case a and Case c (if there are) are combined to infer a U-U relation, and multiple cases of Case b and Case d (if there are) are combined to infer a U-A relation (We refer either of them as a *triangle inference*). The inference probability is up to the number of common neighbors and the types of the relations and the attributes involved. Suppose the target unknown edge is e_{st}, and s, t have a set N_{st} of common neighbors *exactly*, where $N_{st} \subseteq V\backslash\{s, t\}$. The inference probability denoted as $\Pr(e_{st} \mid N_{st})$ is the probability that the relation e_{st} exists given the condition that s, t have all and only the nodes in N_{st} as their common neighbors. We have the following assumption about how the attacker makes inference.

Assumption 1: To infer an unknown edge e_{st}, there are three cases. 1) If s, t have common neighbor(s), then e_{st} exists with a probability (the inference probability); 2) If there is no path connecting s, t, then e_{st} must not exists because there is no evidence indicating it at all; 3) If s, t do not have a common neighbor but there is a path connecting them, the status of e_{st} is TBD because the possibility depends on the path itself.

It is worth mentioning that this assumption also reflects how the real network grows/evolves. For example, two persons are likely to establish friendship if they share some common friends, and a woman could develop a new hobby that her husband likes. Two entities could potentially build a relation if they are indirectly connected by some intermediaries.

We assume that the attacker has already obtained all the necessary inference probabilities by some means, *e.g.* by data mining (how she knows or estimates them is out of the scope of this paper). Most of complex privacy inference types can be easily transformed or decomposed into the base cases, so we assume that the attacker only performs triangle inference.

III. Problem Formulation and Analysis

A. Privacy Inference Definition

Definition 1 (Privacy inference): Given a knowledge graph $G = (V, E, P)$, privacy inference is the problem of computing $p(e_{s,t})$ for any uncertain edge $e_{st} \in E$ where $s \in U$ and $t \in V$, given the inference probabilities $P(e_{st} \mid N_{st})$, $\forall N_{st} \subseteq V\backslash\{s, t\}$. We denote this problem as PI(G, s, t).

Here, the attacker aims to infer a specific kind of information about a target user s, such as a secret connection with somebody $t \in U$ or a private attribute $t \in A$. We assume that privacy inference is limited to a single relation. The attacker can conduct multiple privacy inference steps if she wants to infer more than one relations.

To analyze the problem, we regard G as a random graph in which each edge e is independently *retained* with a probability of its confidence score $p(e)$ or *removed* with a probability of $1-p(e)$. The probability simulates the randomness of success and failure of the event (*i.e.* the existence of the relation) in real life. We flip a biased coin for each edge e in G except e_{st} to determine whether it is retained or removed with a probabilities of $p(e)$ and $1 - p(e)$ respectively, and mark e_{st} as TBD, so we obtain a sample graph G_0, which is referred to as a *conjecture graph*. Take triangle inference as an example, which consists of a target edge e_{st} and some edges connecting from s, t to their common neighbors. If we flip a biased coin at each known edge to remove the edge with a probability of $1 - p(e)$, we obtain a conjecture graph G_0. In this case, if s, t are still connected (via at least a common neighbor) in G_0, then the edge e_{st} exists in this conjecture graph with an inference probability.

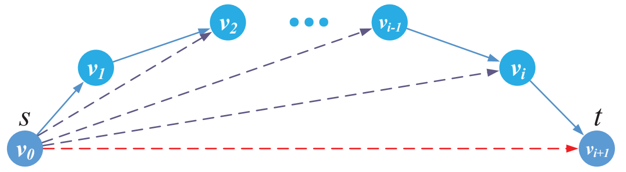


Fig. 2: **A special case of inferring the relation** e_{st}

B. Hardness Proof

For clarification, we first study the special case of privacy inference where all the inference probabilities are set to one. This case is similar to the s-t reliability problem (a.k.a. two-terminal network reliability), which is a classic reliability problem. Given an undirected graph G in which each edge fails independently with a given probability, s-t reliability aims to determine the probability that two nodes $s, t \in E(G)$ remains connected after edge failures. We denote this as REL(G, s, t). Its hardness has been proved in [5], [27].

Theorem 1: REL(G, s, t) is #P-hard.

We will transform the special case to s-t reliability and show its hardness.

Theorem 2: Let G be a knowledge graph, for the special case, PI(G, s, t) is equivalent to REL(G, s, t).

The proof of this Theorem is given in Appendix A. Now we conclude the hardness of privacy inference as follows.

Theorem 3: Privacy inference PI(G, s, t) is #P-hard.

Proof: According to Theorems 2 and 1, the special case of privacy inference is #P-hard and so is privacy inference. ■

IV. Algorithms

For the special case, we can obtain the probability that e_{st} exists by solving the s-t reliability problem via Monte Carlo simulation. Monte Carlo methods are commonly used to approximate the expected value of some random variable that is very hard to compute exactly. They approximate the expected value by taking the empirical mean (sample mean) of independent samples of the variable We flip a biased coin at each edge of G to decide whether to retain or remove the edge, finally get a sample graph G_0, and then check the connectedness of s, t in G_0. If so, we say this sample succeeds; otherwise, it fails. Suppose after sampling m times, there are X samples that succeed, the success ratio $\hat{p} = X/m$ can be used as an estimate of the probability that s, t are connected and thus the $p(e_{st})$.

In a general case, inference probabilities are not necessarily one, so we cannot simply transform it to a reliability problem. We design an iterative algorithm instead to approximate it which is also based on Monte Carlo simulation. In each sample, edge addition will be performed on the knowledge

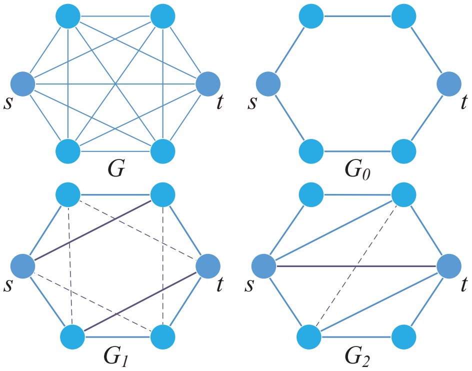


Fig. 3: **A general case of inferring the relation e_{st}:** The attacker first samples a conjecture graph G_0 and then iteratively inferring new edges by triangle inference. The iteration ends in two rounds when e_{st} appears in G_2.

graph for multiple stages, which simulates how real world connections grow gradually. Since the common neighbor sets of node pairs would change after new edges are added, we distinguish them and use $N^i_{v,w}$ to denote the set of common neighbors of v, w we have when the i-th stage is finished.

Stage 0: Given a knowledge graph G and a target unknown relation e_{st}, we generate a conjecture graph G_0 by flipping coins. Then we check if there is a retained path connecting s, t. If not, e_{st} must not exist by Assumption 1 so we can end testing this sample now. Otherwise, the iteration keeps going as follows.

Stage 1: A pair of nodes v, w in G_0 is referred to as a *candidate pair*, if they are unconnected, they share a set $N^0_{v,w}$ of common neighbors, and at least one of them is a user node. (A-A relations are considered to be constant, they do not evolve, and the triangle inference rules do not apply to them, so pairs of non-adjacent attribute nodes are not treated as candidates and will not be connected later.) For each candidate pair, we flip a biased coin so that we add a new edge e_{vw} with a probability of the inference probability $\Pr(e_{vw} \mid N^0_{vw})$, which reflects network evolution. (The influence of a newly added edge on common neighbor set will be handled in the next stage.) Then we update the common neighbor sets and obtain an updated graph G_1 with some new edges, which is the end of the first stage of edge addition.

Stage i: For each candidate pair in G_{i-1}, there are three cases. 1) If v, w did not gain new common neighbor(s) in the $i-1$-th stage, we leave them alone because the coin has been flipped already. 2) If they did not have a common neighbor after stage $i-2$ but now have a set of them $N^{i-1}_{v,w}$, we flip a biased coin to add the new edge e_{vw} with a probability of $\Pr(e_{vw} \mid N^{i-1}_{vw})$. 3) If v, w already shared common neighbor(s) after stage $i-2$ and they gained more after stage $i-1$, we need to re-flip a coin for e_{vw} and cancel the effect caused by the previous failed coin flip(s) for it. We calculate the probability of the coin re-flip p' considering that previous coin flip(s) plus the re-flip should achieve the same effect as a one-time flip, so we have

$$(1 - \Pr(e_{vw} \mid N^{i-2}_{vw}))(1 - p') = 1 - \Pr(e_{vw} \mid N^{i-1}_{vw}), \quad (1)$$

$$p' = \frac{\Pr(e_{vw} \mid N^{i-1}_{vw}) - \Pr(e_{vw} \mid N^{i-2}_{vw})}{1 - \Pr(e_{vw} \mid N^{i-2}_{vw})}. \quad (2)$$

When all the candidate pairs are taken care of, we update the common neighbor sets and obtain an updated graph G_i.

The iteration on one sample ends when either the target edge e_{st} is inferred (the sample succeeds), or no more edges can be added (the sample fails). We sample G_0 and perform the multi-stage edge addition for n times. If there are X samples that succeed, the success ratio $\hat{p} = X/n$ is an estimation of $p(e_{st})$, the probability that e_{st} exists.

Complexity: Let n be sample size, l be the stage number, the time complexity is $O(n|V|^3 l)$. In the worst case where only one edge is added in every stage and e_{st} is the last one added, we have $l = O(|E|) = O(|V|^2)$. Thus, a very loose upper bound of the time complexity is $O(n|V|^5)$.

Error bound: The successful sample number X is a random variable following the binomial distribution, $X \sim B(m, p)$ where p is short for $p(e_{st})$. Therefore, we have

$$\begin{aligned}\Pr(|\hat{p} - p| \le \epsilon) &= \Pr(m(p-\epsilon) \le X \le m(p+\epsilon)) \\ &= \sum_{i=\lceil m(p-\epsilon)\rceil}^{\lfloor m(p+\epsilon)\rfloor} \binom{m}{i} p^i (1-p)^{m-i}\end{aligned} \quad (3)$$

V. Experiment Evaluations

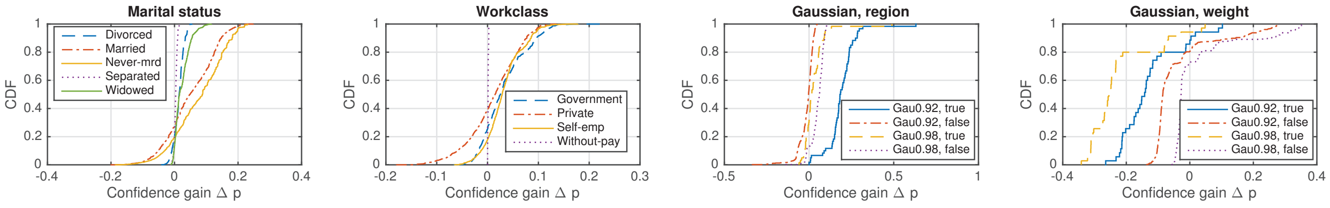
A. Methodology

Data sets: Our simulations use two data sets: UCI Adult [14] and SNAP Pokec social network [33]. Adult is a tabular census data set containing rich information of 30162 individuals. We select 6 attributes: sex, age, race, education, workclass, salary-class. Pokec is a social network data sets from which we select 5 attributes: sex, age, region, height, weight.

Knowledge graph construction: For Adult, we build a knowledge graph for each person with his/her attributes (U-A relations). For Pokec, we randomly select an ego network of 60 users, extract all the involved U-U and U-A relations, and then build a single knowledge graph for these users. A-A relations are calculated by statistics of the data, *e.g.* the ratio of women who have a doctorate degree. All of them are added to the knowledge graphs.

Probability generation: As stated above, the confidence scores of A-A relations are calculated by statistics. The setting of confidence score of a U-U or U-A relation depends on whether the relation is contained in the data. If so, it is a true relation and we set a high probability for it; for a false relation, we set a low probability. We adopt the following methods to generate confidence scores.

1) 1&0: set to 1 or 0 for true and false relations respectively;
2) Uni: generated by the uniform distribution, *e.g.* $U(0.9, 1)$ and $U(0, 0.1)$ for true and false relations respectively (denoted as Uni0.9);



(a) Adult: marital status

(b) Adult: workclass

(c) Pokec: region

(d) Pokec: weight

Fig. 4: **Confidence gain on Adult and Pokec**

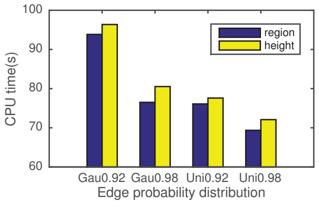


Fig. 5: **CPU time for privacy inference on Pokec**

3) Gau: generated by the Gaussian distribution, *e.g.* $N(0.9, 0.001)$ and $N(0.1, 0.001)$ (denoted as Gau0.9) respectively.

The variances of Gaussian distribution are set by three-sigma rule so that the generated probabilities are within [0,1] at most time (truncated if not). The number of all the inference probabilities $\Pr(e_{vw} \mid N_{vw})$ is $\binom{|V|}{2} \cdot 2^{|V|-2}$. For simplification, we only generate inference probabilities of single common neighbor and aggregate them to calculate those of many common neighbors as follows.

$$\Pr(e_{vw} \mid N_{vw}) = \frac{1}{|N^*_{vw}|} \sum_{i \in N_{vw}} \Pr(e_{vw} \mid \{i\}). \qquad (4)$$

Here N^*_{vw} is the set of potential common neighbors of v and w. The inference probabilities of single common neighbor is generated as follows. For Case d, the inference probabilities are simply set to one because the randomness lies in the attribute dependency. For other three base cases, we generate inference probabilities by statistics.

After building a knowledge graph, we randomly select a target edge and set it to be unknown, then estimate its probability by our algorithm with the sample size set to $n = 10000$. To measure the privacy inference result, we define *confidence gain* as $\Delta p(e_{ij}) = I(e_{ij})(p'(e_{ij}) - p(e_{ij}))$, where e_{ij} is the target edge, p is the initial probability, p' is the updated probability after privacy inference, and $I(e_{ij})$ is a flag indicating whether e_{ij} is true in the ground truth. A positive confidence gain would validate the efficacy of our algorithm.

B. Evaluation on Privacy Inference

Adult: We present the results of privacy inference on the marital status and work-class for each user. Fig. 4(a)(b) displays the confidence gain of privacy inference on true relations, *i.e.* edges that do exist in reality. It is shown that there is a positive confidence gain for at least 60% of the attribute inference tasks in Adult. For some cases like inferring if a person is widowed or if a person works without pay, the ratio of a positive confidence gain reaches 90%. The results are similar for inferring false relations. The run time is less than 1s since the knowledge graph has a small size for each user.

Pokec: We present the results for users' region and weight in Fig. 4(c)(d). Different distributions (Gaussian and uniform) are used for generating edge probabilities. For region, at least 45% of the inference results achieve a positive confidence gain, while the ratio is less than 30% for weight. A possible explanation is that region has more dependence on other knowledge than weight does. For example, people who are friends are more likely to be in the same region. Average CPU time of one privacy inference task is displayed in Fig. 5, which shows that the computation overhead is acceptable.

Discussion: The results shown above demonstrate the feasibility of our model and algorithm. We do not compare with previous proposed algorithms since there is very few related works. The accuracy of privacy inference is greatly up to the accuracy of estimating inference probabilities. As future work, we will utilize deep learning to estimate these probabilities more accurately so as to improve the performance.

VI. Related Work

Privacy attack and protection in relational data has been the focus of research for more than ten years. The most well-known works are k-anonymity [32], l-diversity [18], and t-closeness [16]. Recent work has pointed out the necessity of taking the attacker's prior knowledge into account when estimating the risk of privacy disclosure, *e.g.* [6], [17]. Mostly, these models are limited to the relational data publishing scenario. To avoid the privacy issues in data publishing, some data owners choose not to release the data but allow restricted statistical queries over it [7], [8]. Dwork [8] has shown that no methods can eliminate the possibility of privacy disclosure in statistical query considering that the attacker may have background knowledge.

Privacy inference on network data is also well-studied in the literature, most of which focuses on de-anonymization [21], [31]. They usually consist of two phases: seed identification and mapping propagation. Specifically, the attacker first identifies a few outstanding users as seeds and then propagates the mapping between users and nodes based on their similarities in profiles and structural characteristics. Other work is focused on inferring new personal information with existing network data. For example, Heatherly *et al.* [11] explored how the attacker uses classifiers to learn undisclosed private information of social network users. The most related work by Qian *et al.* [28] used knowledge graphs to model privacy leakage in social network data publishing. They proposed a two-stage attack: de-anonymization and privacy inference, but their analysis of the latter stage is limited. Our work is different in that we formularize privacy inference, show its hardness, and design heuristic algorithms for it.

VII. Conclusion

This paper presents a general model of privacy inference on knowledge graphs. We first model the attacker's background knowledge on knowledge graphs and analyze the base cases of privacy inference. We formally define the problem of privacy inference and prove its #P-hardness. Then we design a heuristic algorithm to estimate it and perform simulations on two real data sets. The experiment results validate the feasibility and effectiveness of our model and algorithm in one sense.

VIII. Acknowledgment

The research of Li is partially supported by NSF CMMI 1436786, NSF CNS 1526638, National Natural Science Foundation of China under Grant No. 61520106007. Key Research Program of Frontier Sciences, CAS. No. QYZDY-SSW-JSC002.

Appendix A
Proof of Theorem 2

Proof: All the inference probabilities are one in the special case, which means for any triangle inference, the edge between two nodes exist iff. they have at least one common neighbor. Given a knowledge graph G, suppose the attacker aims to infer the relation e_{st}. We randomly generate a conjecture graph G_0. Now we prove that e_{st} exists in the G_0 iff. the event succeeds that there is at least one path connecting s, t in G_0 (we denoted this event as $Con(G_0, s, t)$).

Necessity: If $Con(G_0, s, t)$ fails, there is no path connecting s, t, which implies the non-existence of e_{st} by Assumption 1.

Sufficiency: If $Con(G_0, s, t)$ succeeds, suppose the shortest path connecting s, t is s-v_1-v_2-$\cdots$-v_i-t, as depicted in Fig. 2. Let v_0, v_{i+1} be the alias of s, t respectively. By Assumption 1, there is a common neighbor v_1 for s and v_2, so the edge e_{sv_2} exists with an inference probability which is one in the special case. Thus e_{sv_2} can be added to G_0 after applying a triangle inference which simulates network evolution. Likewise, we can infer e_{sv_3} given e_{sv_2} and $e_{v_2v_3}$. We perform triangle inference iteratively to infer $e_{sv_{j+1}}$ given e_{sv_j} and $e_{v_jv_{j+1}}$, $j = 1, 2, \cdots, i$. Finally it can be inferred that e_{st} exists.

Hence, that e_{st} exists in G_0 is equivalent to $Con(G_0, s, t)$. Therefore, the probability that e_{st} exists equals the probability of the connectedness of s, t. In other words, PI(G, s, t) is equivalent to REL(G, s, t). ■

References

[1] Adar, E. User 4xxxxx9: Anonymizing query logs. In *WWW Workshop* (2007).

[2] Aeschlimann, L., Harasgama, R., Kehr, F., Lutz, C., Milanova, V., Müller, S., Strathoff, P., and Tamò, A. Re-setting the stage for privacy: A multi-layered privacy interaction framework and its application. *Mensch und Maschine-Symbiose oder Parasitismus* (2015).

[3] Al Hasan, M., and Zaki, M. J. A survey of link prediction in social networks. In *Social network data analytics*. Springer, 2011, pp. 243–275.

[4] Bian, L., and Holtzman, H. Online friend recommendation through personality matching and collaborative filtering. *UBICOMM* (2011), 230–235.

[5] Colbourn, C. J., and Colbourn, C. *The combinatorics of network reliability*, vol. 200. Oxford University Press New York, 1987.

[6] Du, W., Teng, Z., and Zhu, Z. Privacy-maxent: Integrating background knowledge in privacy quantification. In *SIGMOD* (2008), ACM, pp. 459–472.

[7] Dwork, C. Differential privacy: A survey of results. In *TAMC* (2008), Springer, pp. 1–19.

[8] Dwork, C. Differential privacy. In *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 338–340.

[9] Dwyer, C. Privacy in the age of google and facebook. *IEEE Technology and Society Magazine 30*, 3 (2011), 58–63.

[10] Hakkani-Tür, D., Heck, L., and Tur, G. Using a knowledge graph and query click logs for unsupervised learning of relation detection. In *ICASSP* (2013), IEEE, pp. 8327–8331.

[11] Heatherly, R., Kantarcioglu, M., and Thuraisingham, B. Preventing private information inference attacks on social networks. *TKDE 25*, 8 (2013), 1849–1862.

[12] James, P. Knowledge graphs. *Order 501* (1992), 6439.

[13] Ji, S., Li, W., Srivatsa, M., and Beyah, R. Structural data de-anonymization: Quantification, practice, and implications. In *CCS* (2014), ACM, pp. 1040–1053.

[14] Kohavi, R. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD* (1996), vol. 96, Citeseer, pp. 202–207.

[15] Lange, D., Böhm, C., and Naumann, F. Extracting structured information from wikipedia articles to populate infoboxes. In *CIKM* (2010), ACM, pp. 1661–1664.

[16] Li, N., Li, T., and Venkatasubramanian, S. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE* (2007), IEEE, pp. 106–115.

[17] Li, T., Li, N., and Zhang, J. Modeling and integrating background knowledge in data anonymization. In *ICDE* (2009), IEEE, pp. 6–17.

[18] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. l-diversity: Privacy beyond k-anonymity. *TKDD 1*, 1 (2007), 3.

[19] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. Distant supervision for relation extraction without labeled data. In *IJCNLP* (2009), Association for Computational Linguistics, pp. 1003–1011.

[20] Narayanan, A., and Shmatikov, V. Robust de-anonymization of large sparse datasets. In *S&P* (2008), IEEE, pp. 111–125.

[21] Narayanan, A., and Shmatikov, V. De-anonymizing social networks. In *S&P* (2009), IEEE, pp. 173–187.

[22] Naveed, M., Ayday, E., Clayton, E. W., Fellay, J., Gunter, C. A., Hubaux, J.-P., Malin, B., Wang, X., et al. Privacy and security in the genomic era.

[23] Niu, F., Zhang, C., Ré, C., and Shavlik, J. Elementary: Large-scale knowledge-base construction via machine learning and statistical inference. *IJSWIS 8*, 3 (2012), 42–73.

[24] Niu, F., Zhang, C., Ré, C., and Shavlik, J. W. DeepDive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS 12* (2012), 25–28.

[25] Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, Preprint (2016), 1–20.

[26] Paulheim, H., and Fümkranz, J. Unsupervised generation of data mining features from linked open data. In *WIMS* (2012), ACM, p. 31.

[27] Provan, J. S., and Ball, M. O. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing 12*, 4 (1983), 777–788.

[28] Qian, J., Li, X.-Y., Zhang, C., and Chen, L. De-anonymizing social networks and inferring private attributes using knowledge graphs. In *INFOCOM* (2016), IEEE.

[29] RISTOSKI, P., AND PAULHEIM, H. A comparison of propositionalization strategies for creating features from linked open data. *Linked Data for Knowledge Discovery* (2014), 6.

[30] SOCHER, R., CHEN, D., MANNING, C. D., AND NG, A. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems* (2013), pp. 926–934.

[31] SRIVATSA, M., AND HICKS, M. Deanonymizing mobility traces: Using social network as a side-channel. In *CCS* (2012), ACM, pp. 628–637.

[32] SWEENEY, L. k-anonymity: A model for protecting privacy. *IJUFKS 10*, 05 (2002), 557–570.

[33] TAKAC, L., AND ZABOVSKY, M. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations* (2012), pp. 1–6.

[34] WANG, Y., LI, C., AND CHENG, N. Internet security protection in personal sensitive information. In *CIS* (2014), IEEE, pp. 628–632.

[35] WONG, R. C.-W., FU, A. W.-C., WANG, K., AND PEI, J. Minimality attack in privacy preserving data publishing. In *VLDB* (2007), VLDB Endowment, pp. 543–554.