International Journal of Cooperative Information Systems
Vol. 23, No. 2 (2014) 1441004 (32 pages)

DOI: 10.1142/S0218843014410044

World Scientific
www.worldscientific.com

Privacy-Preserving Assessment of Social Network Data Trustworthiness

Chenyun Dai[*,‡], Fang-Yu Rao[*,§], Traian Marius Truta[†,¶]
and Elisa Bertino[*,‖]

[*]*Department of Computer Science and Cyber Center*
Purdue University, 305 N. University Street
West Lafayette, IN 47907, USA

[†]*Department of Computer Science*
Northern Kentucky University, Nunn Drive
Highland Heights, KY 41099, USA
[‡]*daic@cs.purdue.edu*
[§]*raof@cs.purdue.edu*
[¶]*trutat1@nku.edu*
[‖]*bertino@cs.purdue.edu*

Received 15 February 2013
Accepted 13 May 2014
Published 11 June 2014

Extracting useful knowledge from social network datasets is a challenging problem. While large online social networks such as Facebook and LinkedIn are well known and gather millions of users, small social networks are today becoming increasingly common. Many corporations already use existing social networks to connect to their customers. Seeing the increasing usage of small social networks, such companies will likely start to create in-house online social networks where they will own the data shared by customers. The trustworthiness of these online social networks is essentially important for decision making of those companies. In this paper, our goal is to assess the trustworthiness of local social network data by referencing external social networks. To add to the difficulty of this problem, privacy concerns that exist for many social network datasets have restricted the ability to analyze these networks and consequently to maximize the knowledge that can be extracted from them. This paper addresses this issue by introducing the problem of data trustworthiness in social networks when repositories of anonymized social networks exist that can be used to assess such trustworthiness. Three trust score computation models (absolute, relative, and weighted) that can be instantiated for specific anonymization models are defined and algorithms to calculate these trust scores are developed. Using both real and synthetic social networks, the usefulness of the trust score computation is validated through a series of experiments.

Keywords: Privacy; trustworthiness; social networks.

[§]Corresponding author.

1. Introduction

Social networks have been studied by various research communities for more than 50 years.[1] However, the advent of the online social networks and the wide adoption of such networks by our society have significantly increased the importance of obtaining useful information from these networks. Extracting useful knowledge from social network datasets proves to be a difficult problem and social network mining is currently identified as one of the most challenging problems in data mining research.[2] To add to the difficulty of this problem, privacy concerns exist for many social network datasets. Such concerns have resulted in limited accessibility to social network datasets and thus in reducing the quantity and quality of the knowledge that could be extracted from these datasets. Such knowledge may have important applications, such as disease spreading models in epidemiology, emergency management, protection from cyber-attacks, etc.

While large online social networks such as Facebook and LinkedIn are well known and gather millions of users, small social networks are today becoming increasingly common. Currently, such small niche social networks such as GoFISHn and GoHUNTn are considered as the new trend in online social network usage.[3] Many corporations already use existing social networks to connect to their customers. Seeing the increasing usage of small social networks, such companies will likely start to create in-house online social networks where they will own the data shared by customers. Nowadays, for many services (insurance, airline miles, travel sites, etc.), users have individual accounts on company websites. However, there is no network structure connecting accounts of different users, and therefore the relationships that may exist among such users are not efficiently used by the company. The benefits that can be obtained from adding relationships among customers are significant and include increased possibilities to expand the customer base, increased usage of provided services by the current customers, better marketing opportunities, and so on. Future breakthroughs in social network mining will also expand these opportunities.

Of course, adding relations between their customers has its challenges. A first challenge is to update the existing software and hardware for this new model. Fortunately, such challenge is easily solved once a company allocates the necessary resources. A second challenge is that users must have an incentive to connect among themselves in a company-owned social network. This is not a trivial problem and will likely be a difficult challenge to address. However, by using financial incentives, the users will start to connect to their friends or acquaintances in order to get better deals. For instance, an insurance company may use incentives such as 10% savings on their car insurance costs if a customer registers on his/her social networks site and recommends a minimum number of friends. Next, the amount of savings can be increased based on how many of his/her friends will buy insurance from the same company. Such incentives could also be used to motivate a user to complete his/her profile, and this would allow the insurance company to have a wealth of information about its users that could potentially be used to increase its business. Even more complex models can be created for airline or phone companies. It is also worth noting, that such company-owned social networks are already used by telecommunication companies since they can connect customers based on existing phone calls.

It can be easily seen that such local social networks have many benefits for the organizations that own them. However, the users' main motivation for joining and providing the required information is to get the desired service at a discount price or any other incentive associated with the use of this company-owned social networking site. Therefore, it is expected that users will be less likely to provide only accurate information in their profiles (due to privacy concerns or because of other advantages that could be obtained by partially faking profile information). An example of a possible advantage that could be obtained is as follows. A user can report his marital status as single although he is married. The reason of such reporting is that his wife may be under 25 years old and adding her in the profile may result in the insurance agency include her in the insurance policy and therefore in increasing the auto insurance rate. Other examples include misreporting of address, age, and so on. However, due to incentives and since relationships are approved by both participants, links in such social network will likely to be accurate.

This possibility of faking part of profile data will diminish the utility of the data. The organizations that own such data will benefit from it if they can assess the trustworthiness of such data and if they can identify possible fake information. Unfortunately, due to privacy regulations, large social network datasets that could potentially be used to verify local information are not available in clear due to privacy concerns. However, we can expect that anonymized social network datasets be available, perhaps upon payment, and they can be used to determine the trustworthiness of local data. The existence of significant amount of work in data privacy and in particular social network privacy and anonymity would make it possible to disclose this data in anonymized form.

To summarize, in our framework, one company will create and maintain a local social network (usually of its customers). The company also has access to one or more anonymized social network datasets that contain the nodes (customers) present in its local social network. Based on these anonymized networks, the company will compute a trust score of its customers and based on these score values will decide to make additional verifications regarding the validity of the reported data or take other actions.

The main contributions of this paper are as follows:

- To introduce the problem of data trustworthiness in social networks when repositories of anonymized social networks exist. To our knowledge there is no prior work that addresses data trustworthiness in social networks.
- To present three trust score computation models (absolute, relative, and weighted) that can be instantiated for specific anonymization models.
- To introduce algorithms to calculate the trust score for nodes (customers) from the local social network for two existing social network anonymization models.

- To illustrate the validity of our trust score computation through experiments. We used both real and synthetic social networks in our experiments.

The remaining of the paper is structured as follows. Section 2 discusses related work. Sections 3 and 4 define the problem and introduce the notion of trust score for customer profile information, respectively. Section 5 summarizes the social network anonymization model introduced by Campan and Truta[4] and presents an algorithm for computing the trust score when such an anonymized social network is available. Section 6 discusses the social network anonymization model by Liu and Terzi.[5] A similar algorithm to compute the trust score is also introduced in this section. Experimental results on real and synthetic social network datasets are reported in Sec. 7. Section 8 concludes the paper and outlines future research directions.

2. Related Work

Computing trust for social networks based on external anonymized networks to our knowledge has not yet been investigated. The closest areas to our work are *social network privacy* and *trust* (*reputation*) *in social networks*. Other areas that are of interest include *graph isomorphism/approximate graph matching* and *modeling social networks with node attributes*.

Social network privacy is a growing area by itself and there are a large number of papers in this field. Many recent studies including the two anonymization models used in this paper focus on protecting vertex identities in published social networks.[4–11] Cheng *et al.*[6] identify two kinds of information, which describe node information and link information, as main types of sensitive information. They show that the problem to protect such information is NP-hard and propose a solution based on k-isomorphism. Tai *et al.*[8] formulate a new privacy approach for protecting the community identity of each individual in published social networks. To ensure privacy for community identities, they propose k-structural diversity which requires that, for each vertex, there are other vertices with the same degree locating in atleast $k-1$ other communities. Wu *et al.*[9] propose the k-symmetry model to achieve privacy for social network. Their main idea is to modify the network so that for each vertex v, there exist atleast $k-1$ other vertices such that each of them is an image of v under some automorphism of the modified network. Zhou *et al.*[11] study one type of privacy attacks in social networks: neighborhood attacks, in which the adversary has the knowledge of neighborhood information of the victim. Most of these approaches focus on a k-anonymity like approach, while others use a randomization approach.[10] A different privacy model is proposed by Gehrke *et al.*[12] defined as an extension of differential privacy.[13] The focus of such model is on releasing information about the structure of the social network instead of an anonymized network. De-anonymizing users in social networks is also analyzed in several papers.[7,14,15] Backstrom *et al.* introduce passive and active attacks for de-anonymization of social networks.[14] Narayanan and Smatikov propose an algorithm that using the social network topologies aims at de-anonymizing the social network.[15] Hay *et al.*[7] use structural queries such as subgraph and hub fingerprint queries to de-anonymize social networks.

Trust in social networks is in general considered at two levels: *global* and *local.*[16] At the global level, the trustworthiness of each node is computed based upon complete graph information, while at the local level, trust is computed with respect to the perspective of each specific user. Local trust models include a reputation system based on maximum flows,[17] a system based on weighted paths,[18] and a system based on spreading activation.[16] Global trust is in general computed based on social ties and high trust is typically associated with influential/authoritative nodes in the network. Methods such as Katz's index,[19] PageRank,[20] and HITS[21] are used to find such relevant nodes in a social network. Some recently developed websites[22,23] also use the same idea of reputation systems to detect and remove fake profiles from social networks. To our knowledge, there is no global trust computation approach for a local social network that is owned by an organization as in the problem that we address in this paper.

Graph isomorphism and exact subgraph matching are well-known NP-problems.[24,25] While improvements to the original backtracking algorithm proposed by Ullman[25] exist, they are not efficient for large networks. Cordella *et al.*[26] propose a deterministic matching method for verifying both isomorphism and subgraph isomorphism. A state space representation (SSR) of the matching process is used and a set of five feasibility rules for pruning the search tree are introduced in their paper. The main improvement introduced in their work is that the data structures employed during the exploration of the search space are organized in such a way to significantly reduce memory requirements. However, the algorithm proposed is only suitable for matching graphs with thousands of nodes and edges, which is much smaller compared to the size of online social networks. A more efficient approximate graph matching methods exist.[27,28] Gori *et al.*[27] proposed a graph matching algorithm which uses the random walk model to compute topological features that can be used to reconstruct the mapping between corresponding nodes in the isomorphic graphs. The features and the matching can be computed efficiently. Pedarsani *et al.*[28] study the problem of finding the correct mapping between the node sets of two structurally similar graphs. Using ideas from graph sampling in modeling evolution of networks, they proposed a probabilistic model for deriving two sampled versions of an underlying graph as "noisy" versions of the networks to be matched. They prove that using the simplest matching criterion based only on network topology, a perfect matching between the nodes can be established with high probability as the network size grows large, under simple conditions for the sampling process. Unfortunately, they try to minimize node mismatches[27] or edge mismatches[28] and they are not applicable to our trust score computation.

Modeling social networks with node attributes is a very new field in social network.[29,30] Mislove *et al.* use node attribute values as well as the entire graph structure to predict the attribute's values of the remaining nodes.[29] Kim and Leskovec present a multiplicative attribute graph model that considers nodes with categorical attributes and models the probability of edges forming between different nodes.[30] Zheleva *et al.*[31] attempt to predict the private attributes of users in real-world datasets. In addition to using general relational classification, they introduce a group based classification by taking advantage of specific types of attributes in each of the social networks. Heatherly *et al.*[32] propose two learning models based on node properties and friendship links. In addition, they explore the efiect of removing node properties and links in preventing sensitive information leakage. These approaches have the potential of being used to generate synthetic social network with user's profiles that follow the same properties as the real-world networks.

3. Problem Definition

We assume that a company has created its own social network. Since this network is usually obtained from its own customers that willingly share their data with the company, we call such a company *data owner*. We use the term *local social network* to refer to the company-owned network. We model this local social network as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges. Each node represents an individual entity such as a customer and each edge represents an existing relation between two nodes. Each node has an associated profile represented by a set of attributes. This set of attribute contains *identifier*, *quasi-identifier*, and occasionally *sensitive* attributes[4] that are supposed to be known by the data owner. We assume that all relationships in this local social network are binary. Moreover, we assume all relationships to be of the same type and, as a result, we represent them via unlabeled undirected edges. We use X or Y to represent individual nodes, and X_i, $i = 1, \ldots, n$, to represent all the nodes in $\mathcal{N}$, where $n = |\mathcal{N}|$. We use the notation $X.A$ to refer to the attribute A's value for the node X.

We assume that the owner of the local social network has access to one or more anonymized social networks. An anonymized social network is provided by an external organization (such as Facebook or LinkedIn) that protects the identity and the sensitive information in the social network data by using an anonymization process. We will discuss two existing social network anonymization models in Secs. 5 and 6. We assume that there are s such anonymized social networks available. We represent these networks as $\mathcal{AG}_j = (\mathcal{AN}_j, \mathcal{AE}_j)$ $(j = 1, s)$. Each such anonymized social network is created by the external organization, owner of the social network, from an original graph. We label the corresponding original graphs as $\mathcal{G}_j = (\mathcal{N}_j, \mathcal{E}_j)$. It is worth noting that these graphs are large compared to the local social network.

We initially assume that each anonymized social network contains all nodes from $\mathcal{N}$. Moreover, we assume that all edges from the local social network are present in the underlying social networks from which the anonymized social networks were created. In other words, $\mathcal{N} \subseteq \mathcal{N}_j$ and $\mathcal{E} \subseteq \mathcal{E}_j$, for all $j = 1, \ldots, s$. While both of those assumptions seem to restrict the usefulness of this model, they help us to fully define the problem while focusing on the trustworthiness of the local social network.

In addition to the above assumptions, we initially consider that the anonymized social networks are fully trusted (no fake information was provided to them by the corresponding users). We relax this assumption in Sec. 7 by introducing trust scores associated with nodes of these anonymized networks.

Since individuals from the local social network $\mathcal{G}$ are present in each anonymized social networks $\mathcal{AG}_j$, the set of attributes from $\mathcal{N}$ and the set of attributes from $\mathcal{AN}_j$ are not disjoint. However, due to anonymization, all identifier attributes are removed from $\mathcal{AN}_j$. Also, the sensitive attributes are not shared between $\mathcal{N}$ and $\mathcal{AN}_j$ (due to privacy concerns and/or regulations the sensitive data that might be owned by the owner of the local social network is not available to the external organization that creates the anonymized networks). Based on these assumptions, only a subset of quasi-identifier attributes is common.

We also assume that the owner of the local social network trusts the validity of some of the attributes shared with the anonymized networks, while consider other attributes less reliable.

The data owner wants to determine a trust score of nodes information with respect to those non-trusted attributes and to perform additional verification if this score is low. As we discuss later, the data owner will compare the trust scores for all nodes and will select a percentage of the lowest scores for this additional verification. Without limiting the applicability of our approach, we assume that we have only one target attribute, labeled B, which may contain misreported/non-trusted information. When more attributes are non-trusted, we can compute the trust score for one attribute at a time. In order to assess the trustworthiness of values for this attribute B, the attribute must exist in each anonymized social networks (otherwise the anonymized social network is not useful and will not be considered). We denote the other trusted attributes existing in both the anonymized social network and the local social network with $A_1, A_2, \ldots, A_q$, where q is the number of shared attributes. Note that this set depends on the selected anonymized social network; in other words, the set of common attributes is specific for each anonymized social network and changes for each selected anonymized social network (the number q also changes). Since in our analysis we use only one anonymized social network at the time (see Fig. 1), for simplicity, we use the same notation for each set of common attributes between local and anonymized social network.

4. Trust Scores

We use the notation $TS(X.B)$ (*trust score*) to denote the trustworthiness for the attribute B's value for the node X. For this measure we use all available s anonymized social networks. To obtain this measure, we use the *intermediary trust scores* that we compute for each anonymized social network. We use the notation $TS_j(X.B)$ $(j = 1, \ldots, s)$ when $\mathcal{AG}_j$ is used in this intermediary measure.

We compute such an intermediary trust score, $TS_j(X.B)$, by matching a node X from the local social network to nodes from an anonymized social network. We

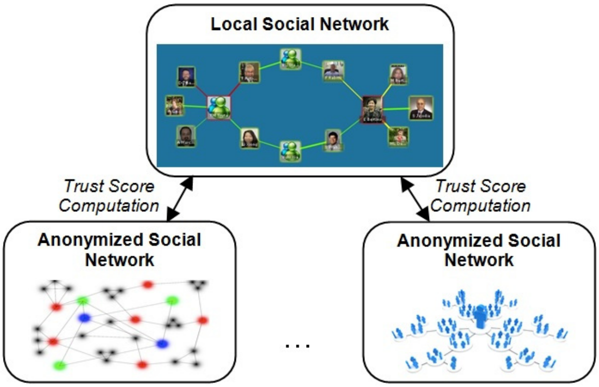


Fig. 1. Trust score computation framework.

consider in this matching, the node attribute's information (that is, the values of attributes B, $A_1, \ldots, A_q$) and the graph structure. The approach used to compute such score is not unique. For instance, we can consider the trust score as the percentage of nodes from the anonymized social network that could potentially be X. We refer to this approach as *absolute trust score*. A second approach first computes how many nodes from the anonymized network can be X when only the trusted attributes $A_1, \ldots, A_q$ and the graph structure are used. We then find the subset of those nodes that match the value of the B attribute as well (note that a non-generalized value will match its ancestors on the value generalization hierarchy). The number of those nodes divided by the number of nodes that matches X based only on trusted values and graph structure is our second measure of trust. We refer to this measure as *relative trust score*.

Our last approach to compute an intermediary trust score includes a weight that depends on how the values of attribute B are published in the anonymized social network. In most anonymized networks, generalization[33,34] is used to anonymize the quasi-identifier attributes, and in this case we would like to differentiate between cases when a specific value (such as the exact name of a city) or a generalized value (such as the name of the country) is used. We thus extend the relative trust score computation approach by assigning a higher weight to matches of X with anonymized nodes that contain more specific information for B. More precisely, the weight associated with a specific value is 1, and the weight decreases when the amount of generalization increases. For example, considering the attribute *city*, the weight associated with a single value like *Chicago* is 1 and the weight associated with a generalized value like *Illinois* is 1/10 assuming that there are 10 cities in Illinois in the value generalization hierarchy used for this attribute. We assume each weight to be a strictly positive value. We refer to this approach as *weighted trust score*.

Before we formally represent these three measures in Definitions 1–3, we introduce the following notations:

- n_j: the number of nodes in the anonymized graph $\mathcal{AG}_j$.

- *matched_B*$(X, \mathcal{AG}_j)$: the number of nodes in $\mathcal{AG}_j$ that can potentially be X. All attributes $(B, A_1, \ldots, A_q)$ and the graph structure are used in this determination.
- *matched_No_B*$(X, \mathcal{AG}_j)$: the number of nodes in $\mathcal{AG}_j$ that can potentially be X. Only trusted attributes $(A_1, \ldots, A_q)$ and the graph structure are used in this determination.
- *matched_weighted_B*$(X, \mathcal{AG}_j)$: each node in $\mathcal{AG}_j$ that can potentially be X will contribute with a weight in the interval $(0, 1]$. This weight is based on the amount of generalization for the attribute B's value in the anonymized graph (see previous paragraphs for a discussion of weights). All such weights are added for the final result. All attributes $(B, A_1, \ldots, A_q)$ and the graph structure are used in this determination. It can be easily noticed that for any X, *matched_weighted_B*$(X, \mathcal{AG}_j) \leq$ *matched_B*$(X, \mathcal{AG}_j)$.

Definition 1 (Intermediary Absolute Trust Score). The intermediary absolute trust score for a node X with respect to an anonymized network $\mathcal{AG}_j$, denoted by $ATS_j(X.B)$, is defined as:

$$ATS_j(X.B) = \frac{matched\_B(X, AG_j)}{n_j}.$$

Definition 2 (Intermediary Relative Trust Score). The intermediary relative trust score for a node X with respect to an anonymized network $\mathcal{AG}_j$, denoted by $RTS_j(X.B)$, is defined as:

$$RTS_j(X.B) = \frac{matched\_B(X, AG_j)}{matched\_No\_B(X, AG_j)}.$$

Definition 3 (Intermediary Weighted Trust Score). The intermediary weighted trust score for a node X with respect to an anonymized network $\mathcal{AG}_j$, denoted by $WTS_j(X.B)$, is defined as:

$$WTS_j(X.B) = \frac{matched\_weighted\_B(X, AG_j)}{matched\_No\_B(X, AG_j)}.$$

The techniques for computing those three intermediary trust score measures depend on the specific techniques used for the social network anonymization, and we defer their presentation to Secs. 5 and 6.

The range for any such intermediary trust score measure is between 0 and 1. The value of 0 is obtained when X does not match any node from the anonymized social network. The value of 1 is obtained in different situations depending on the used intermediary trust score measure. For the absolute trust score, the value of 1 means that all nodes in the anonymized network can potentially be X. For the relative trust score, a value of 1 is obtained when the use of the B attribute value will not limit the number of matches the node X has in the anonymized graph. For the weighted trust score, the value of 1 is obtained when the relative trust score measure is 1 and the attribute B has specific values (non-generalized) in each matched node in the anonymized social network.

Definition 4 (Trust Score). The trust score for a node X with respect to a non-trusted quasi-identifier attribute B value is defined as the average of intermediary trust score values computed for all anonymized social networks if all such values are strictly greater than zero and 0 if one such intermediary value is 0. Let s be the number of available anonymized social networks; the trust score is defined as:

$$TS(X.B) = \begin{cases} 0, & \text{if } \exists\, j \in 1, \dots, s \text{ such that } TS_j(X.B) = 0 \\ \operatorname*{avg}_{j=1,\dots,s} TS_j(X.B), & \text{otherwise.} \end{cases}$$

In Definition 4, we can use any intermediary trust score measure (absolute, relative, and weighted) and consequently we will obtain three different measures for the total trust score. We use $\mathit{ATS}(X.B)$, $\mathit{RTS}(X.B)$, and $\mathit{WTS}(X.B)$ to denote those three total trust score measures.

The following properties hold.

Property 1. $\mathit{ATS}(X.B)$, $\mathit{RTS}(X.B)$, and $\mathit{WTS}(X.B)$ take values in the interval $[0, 1]$ for all nodes X.

Proof. Since all intermediary trust score values are between 0 and 1, their average is also between 0 and 1. □

Property 2. If any trust score measure is 0 for a given node X, then the other two trust score values are also 0. In other words,

(a) If $\mathit{ATS}(X.B)$ is equal to 0, then $\mathit{RTS}(X.B)$ and $\mathit{WTS}(X.B)$ are 0.
(b) If $\mathit{RTS}(X.B)$ is equal to 0, then $\mathit{ATS}(X.B)$ and $\mathit{WTS}(X.B)$ are 0.
(c) If $\mathit{WTS}(X.B)$ is equal to 0, then $\mathit{ATS}(X.B)$ and $\mathit{RTS}(X.B)$ are 0.

Proof. (a) This follows directly from the definitions of trust scores. If $\mathit{ATS}(X.B)$ is 0 then atleast in one anonymized network AG_j, $\mathit{ATS}_j(X.B)$ is 0. Then based on Definition 1, $\mathit{matched\_B}(X, \mathcal{AG}_j)$ is 0. Since the same numerator is used for the intermediary relative trust score, $\mathit{RTS}_j(X.B)$ is also 0, and consequently $\mathit{RTS}(X.B) = 0$. We also know that $\mathit{matched\_weighted\_B}(X, \mathcal{AG}_j) \leq \mathit{matched\_B}(X, \mathcal{AG}_j)$, and, therefore, $\mathit{matched\_weighted\_B}(X, \mathcal{AG}_j)$ is 0. Based on this and Definitions 3 and 4, $\mathit{WTS}(X.B) = 0$. The proofs for (b) and (c) are similar. □

Property 3. For any node X, $\mathit{ATS}(X.B) \leq \mathit{RTS}(X.B)$.

Proof. We already know (Property 2) that if one trust score is 0 the other one is also 0; in this case Property 3 holds. For each anonymized social network

$$\begin{aligned} &\mathcal{AG}_j, j = 1, \dots, s, \\ \mathit{ATS}_j(X.B) &= \frac{\mathit{matched\_B}(X, AG_j)}{n_j} \leq \frac{\mathit{matched\_B}(X, AG_j)}{\mathit{matched\_No\_B}(X, AG_j)} \\ &= \mathit{RTS}_j(X.B) \end{aligned}$$

($matched\_No\_B(X, \mathcal{AG}_j)$ will be at most equal to the number of nodes from $\mathcal{AG}_j(n_j)$). By computing the average of intermediary trust scores we get $ATS(X.B) \leq RTS(X.B)$. □

Property 4. For any node X, $WTS(X.B) \leq RTS(X.B)$.

Proof. Because of Property 2, we know that if one trust score is 0, the other one is also 0. In this case, Property 4 holds. For any anonymized social network

$$\begin{aligned} &\mathcal{AG}_j, j = 1, \ldots, s, \\ WTS_j(X.B) &= \frac{matched\_weighted\_B(X, AG_j)}{matched\_No\_B(X, AG_j)} \\ &\leq \frac{matched\_B(X, AG_j)}{matched\_No\_B(X, AG_j)} = RTS_j(X.B). \end{aligned}$$

By computing the average of intermediary trust scores, we get $WTS\ (X.B) \leq RTS(X.B)$. □

The data owner computes the trust score for local nodes, either for all nodes or for a subset that it has already identified through other means as being less trusted with respect to their self-reported B attribute value. Of particular importance are nodes with a trust score of 0. Based on our problem assumptions, such nodes have fake values for the attribute B. The data owner can extract the set of possible values for the attribute B that such a node X may have from each anonymized social network. This is executed by considering all B attribute values from all the nodes from each anonymized network that match a node X when only trusted attributes $A_1, \ldots, A_q$ and the graph structure are used. Therefore a set of possible values for attribute B is obtained for each anonymized network. The last step in this procedure is to intersect all these sets of possible values. Again, due to the assumption that the underlying graphs contain only valid information, the set of possible B attribute values obtained by such intersection will never be empty. For those nodes that have a positive trust score, the data owner is in general not certain of the correctness of the B attribute value. The strategy for those nodes is to sort them based on trust score values and to consider a predefined number (or percentage) of nodes that have the lowest trust score values for a human analysis. These nodes have definitely a higher risk of misreported information. In Sec. 7, we apply this strategy for each of the three trust score measures and we experimentally compare which trust score provides the most accurate results.

The weighted trust score measure has an interesting property. If the intermediary weighted trust score for a node X and an anonymized network $\mathcal{AG}_j$ is 1, then $matched\_No\_B(X, \mathcal{AG}_j)$ is equal to $matched\_weighted\_B(X, \mathcal{AG}_j)$. This means that the set of nodes from $\mathcal{AG}_j$ that matches X is the same whether the attribute B value of the node X is used or not. Moreover for each node from this set, the corresponding attribute B value must be identical to the attribute B value of node X because if this value would be generalized the weight associated with the computation of $matched\_weighted\_B(X, \mathcal{AG}_j)$ would be strictly less than 1 and the intermediary weighted trust score could not be 1. This means that the data owner knows with absolute certainty that the attribute B value for node X is correct, and thus node X is fully trustworthy. It is easy to notice that the other two trust measures do not have this interesting property. This observation leads us to the updated version of weighted trust score definition (see Definition 5).

Definition 5 (Weighted Trust Score). The weighted trust score for a node X with respect to a non-trusted quasi-identifier attribute B value is defined as follows:

$$WTS(X.B) = \begin{cases} 0, & \text{if } \exists\, j \text{ in } 1, \ldots, s \text{ such that } WTS_j(X.B) = 0 \\ 1, & \text{if } \exists\, j \text{ in } 1, \ldots, s \text{ such that } WTS_j(X.B) = 1 \\ \underset{j=1,\ldots,s}{\operatorname{avg}} WTS_j(X.B), & \text{otherwise.} \end{cases}$$

5. Trust Scores for *k*-Anonymous Clustered Social Network

5.1. *Anonymization model*

As our approach depends on the specific anonymization approach used, we first succinctly present the social network anonymization model introduced in Ref. 4.

Consider an initial social network to be anonymized. Using a grouping strategy, one can partition the nodes from this network into pairwise disjoint clusters. The goal is to make sure that any two nodes from any cluster are indistinguishable based on their relationships and quasi-identifier attributes values. To achieve such goal, Campan and Truta developed intra-cluster and inter-cluster edge generalization techniques that were used for generalizing the social network structure. They also used generalization[33,34] for quasi-identifier attributes values and each cluster will have its profile replaced by the generalization information of that cluster (the minimal covering tuple for that cluster). This generalization information is defined next.

Definition 6 (Generalization Information of a Cluster). Let $cl = \{X_1, X_2, \ldots, X_u\}$ be a cluster of tuples corresponding to nodes selected from $\mathcal{IN}$, $\mathcal{QN}$ be the set of numerical quasi-identifier attributes, and $\mathcal{QC}$ be the set of categorical quasi-identifier attributes. The generalization information of cl w.r.t. quasi-identifier attribute set $\mathcal{QI} = \mathcal{QN} \cup \mathcal{QC}$ is the "tuple" $gen(cl)$, having the schema $\mathcal{QI}$, where:

- For each categorical attribute $C \in \mathcal{QI}$, $gen(cl)[C]$ is equal to the lowest common ancestor in $\mathcal{H}_{Cj}$ of $\{X_1.C, \ldots, X_u.C\}$. We denote by $\mathcal{H}_C$ the hierarchies (domain and value) associated with the categorical quasi-identifier attribute C.

- For each numerical attribute $N \in \mathcal{QI}, gen(cl)[N]$ is equal to the interval $[\min\{X_1.N, \ldots, X_u.N\}, \max\{X_1.N, \ldots, X_u.N\}]$.

For a cluster cl, its generalization information $gen(cl)$ is the tuple having as value for each quasi-identifier attribute, numerical or categorical, the most specific common generalized value for all that attribute values from cl tuples. In an anonymized graph, each tuple from cluster cl will have its quasi-identifier attributes values replaced by $gen(cl)$.

The notion of k-anonymous anonymized social network is fully specified in the following two definitions.

Definition 7 (Anonymized Social Network). Given an initial social network, modeled as a graph $\mathcal{IG} = (\mathcal{IN}, \mathcal{IE})$, and a partition $\mathcal{S} = \{cl_1, cl_2, \ldots, cl_v\}$ of the nodes set $\mathcal{IN}$, $\bigcup_{j=1}^{v} cl_j = \mathcal{IN}$; $cl_i \cap cl_j = \emptyset$; i, $j \in \{1, 2, \ldots, v\}$, $i \neq j$; the corresponding anonymized social network $\mathcal{AG}$ is defined as $\mathcal{AG} = (\mathcal{AN}, \mathcal{AE})$, where:

- $\mathcal{AN} = \{Cl_1, Cl_2, \ldots, Cl_v\}$, Cl_j is a node corresponding to the cluster $cl_j \in \mathcal{S}$ and is described by the "tuple" $gen\ (cl_j)$ (the generalization information of cl_j, w.r.t. quasi-identifiers) and the intra-cluster generalization pair $(|cl_j|, |\mathcal{IE}_{clj}|)$;
- $\mathcal{AE} \subseteq \mathcal{AN} \times \mathcal{AN}$; $(Cl_i, Cl_j) \in \mathcal{AE}$ if Cl_i, $Cl_j \in \mathcal{AN}$ and $\exists\, X \in cl_j$, $Y \in cl_j$, such that $(X, Y) \in \mathcal{IE}$. Each edge $(Cl_i,\ Cl_j) \in \mathcal{AE}$ is labeled with the inter-cluster generalization value $|\mathcal{IE}_{cli,clj}|$.

By construction, all nodes from a cluster cl collapsed into the generalized node Cl are indistinguishable from each other.

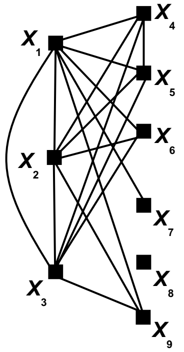
In order to satisfy the k-anonymity property for an anonymized social network each cluster from partition $\mathcal{S}$ must have the size at least k.

Definition 8 (K-Anonymous Social Network). An anonymized social network $\mathcal{AG} = (\mathcal{AN}, \mathcal{AE})$, where $\mathcal{AN} = \{Cl_1, Cl_2, \ldots, Cl_v\}$, and $Cl_j = [(|cl_j|, |\mathcal{IE}_{clj}|)]$, $j = 1, \ldots, v$ is k-anonymous if $|cl_j| \geq k$ for all $j = 1, \ldots, v$.

Campan and Truta developed a social network anonymization algorithm called *Sangreea*, which creates one cluster at a time. In each new formed cluster, nodes are included that are as similar as possible, both in terms of their quasi-identifier attribute values, and in terms of their neighborhood structure. For complete details of this algorithm please refer to Ref. 4.

Next, we present an example of application of this anonymization model. This example will be continued in Sec. 5.2 and Sec. 6 to illustrate the trust score computation.

Example 1. Consider the social network $\mathcal{IG}_1$ in Fig. 2. $\mathcal{IG}_1$ contains nine nodes, described by the quasi-identifier attribute: *sex*, *age*, and *city*. The *age* quasi-identifier is numerical, *sex* and *city* are categorical. The attribute *sex* can take two values (M and F) and these values can be generalized only to person (P). For this dataset, the attribute *city* takes only four values (*Chicago*, *Detroit*, *Miami*,



	Sex	Age	City
X_1	M	20	Detroit
X_2	F	25	Detroit
X_3	F	30	Detroit
X_4	M	24	Chicago
X_5	M	32	Chicago
X_6	F	35	Miami
X_7	F	33	Seattle
X_8	F	40	Detroit
X_9	F	45	Detroit

Fig. 2. The social network $\mathcal{IG}_1$.

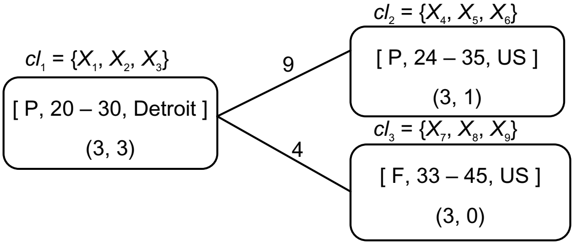


Fig. 3. The three-anonymous social network $\mathcal{AG}_1$.

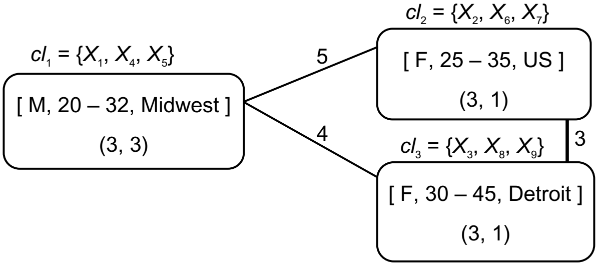


Fig. 4. The three-anonymous social network $\mathcal{AG}_2$.

and *Seattle*). The cities *Chicago* and *Detroit* have the value *Midwest* as their direct ancestor in the value generalization hierarchy. *Miami*, *Seattle*, and *Midwest* have *US* as their ancestor.

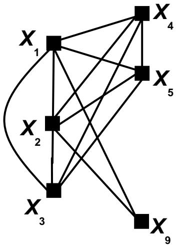
In Figs. 3 and 4, two three-anonymous social networks, $\mathcal{AG}_1$ and $\mathcal{AG}_2$, are illustrated. For generating the anonymized network in Fig. 3, the structure was considered more important in the determination of clusters, while for the anonymized network in Fig. 4, the quasi-identifiers attribute values were given priority. In each cluster, we represent the generalization information of that cluster, followed by a pair of numbers that represent the number of nodes and the number of intra-cluster edges. Outside each cluster, we represent the set of original nodes that were clustered together. The edges' weight represents the number of edges between nodes from the two connected clusters.

5.2. *Intermediary trust scores computations*

In this section, we present a method to compute the intermediary trust scores when the k-anonymous clustered model is used for anonymizing the social networks. We explain our approach through the following example which continues Example 1 from the previous section.

Example 1 — Cont. Suppose that the data owner has a local social network $\mathcal{G}_1$ shown in Fig. 5. The data owner has access to both k-anonymous social networks $\mathcal{AG}_1$ and $\mathcal{AG}_2$ in Figs. 3 and 4. The data owner fully trusts the values of the *age* and *sex* attributes, and wants to assess the trustworthiness of the *city* attribute. Note that all assumptions made in Sec. 3 are satisfied. Suppose now that three nodes in $\mathcal{G}_1$ contain fake information for the attribute *city*. These fake values are underlined and italicized in Fig. 5. The data owner does not have access to $\mathcal{IG}_1$, and nodes labels are used just for illustration purposes.

In order to compute the trust scores, the first step is to match each node from the local graph to clusters from each anonymized graph. We perform two types of matches. First, we use the graph structure and the *sex* and *age* attributes values. Second, we also add the non-trusted attribute *city* to this match process. Algorithms for these procedures are detailed later in this section. After these matches, we can directly compute the intermediary absolute and relative trust scores according to Definitions 1 and 2. For the weighted trust score, we define a set of weights as follows: for a single value in a cluster the weight is 1; for *Midwest*, the weight is 1/2 (there are two possible values, Detroit or Chicago); for US the weight is 1/4 (again, due to four possible *city* values). Now we have all the tools to compute the intermediary trust scores. Tables 1 and 2 show the intermediary trust score computations with respect to $\mathcal{AG}_1$ and $\mathcal{AG}_2$ respectively. Table 3 shows the results of the combined trust scores computed according to Definitions 4 (for absolute and relative trust scores) and 5 (for weighted trust score).



	Sex	Age	City
X_1	M	20	*Chicago*
X_2	F	25	Detroit
X_3	F	30	*Miami*
X_4	M	24	Chicago
X_5	M	32	Chicago
X_9	F	45	*Seattle*

Fig. 5. The local social network $\mathcal{G}_1$.

Table 1. The intermediary trust scores for $\mathcal{AG}_1$.

	Clusters Matched Using City	Clusters Matched no City	*matched_B*	*matched_No_B*	*matched_weighted_B*	ATS_1	RTS_1	WTS_1
X_1	∅	cl_1	0	3	0	0	0	0
X_2	cl_1	cl_1	3	3	3	1/3	1	1
X_3	cl_2	cl_1, cl_2	3	6	3/4	1/3	1/2	1/8
X_4	cl_2	cl_1, cl_2	3	6	3/4	1/3	1/2	1/8
X_5	cl_2	cl_2	3	3	3/4	1/3	1	1/4
X_9	cl_3	cl_3	3	3	3/4	1/3	1	1/4

Table 2. The intermediary trust scores for $\mathcal{AG}_2$.

	Clusters Matched Using City	Clusters Matched no City	*matched_B*	*matched_No_B*	*matched_weighted_B*	ATS_2	RTS_2	WTS_2
X_1	cl_1	cl_1	3	3	3/2	1/3	1	1/2
X_2	cl_2	cl_2	3	3	3/4	1/3	1	1/4
X_3	cl_2	cl_2, cl_3	3	6	3/4	1/3	1/2	1/8
X_4	cl_1	cl_1	3	3	3/2	1/3	1	1/2
X_5	cl_1	cl_1	3	3	3/2	1/3	1	1/2
X_9	∅	cl_3	0	3	0	0	0	0

Table 3. The trust scores for nodes from $\mathcal{G}_1$.

	ATS	*RTS*	*WTS*
X_1	*0*	*0*	*0*
X_2	1/3	1	5/8
X_3	*1/3*	*1/2*	*1/8*
X_4	1/3	3/4	5/16
X_5	1/3	1	3/4
X_9	*0*	*0*	*0*

For two nodes (X_1 and X_9) the trust score is 0; based on this trust score the data owner is able to determine that the *city* attribute values reported by these two nodes are not correct. However, for X_3, the trust scores values are positive, and the data owner does not know for certain if the values are correct or not. The data owner will use the strict positive values as follows. It will order them increasingly and will consider the ones with smaller values for additional verification. This additional analysis will involve human intervention. The data owner may select a specified percentage of values for this additional verification, or may consider a trust score value as a threshold. It is important to notice that the trust score values are relative to each other and the availability of more anonymized networks as well as the quality of those anonymized networks have a major impact on the values of those trust scores. In our example, the weighted and relative trust scores for X_3 are the lowest among all the strictly positive values, thus this node will be the first to be investigated by the data owner. We assess the accuracy of all trust score measures in Sec. 7.

We now introduce an algorithm for *matched_No_B*$(X, \mathcal{AG})$ that we used in our trust score computations.

```
Algorithm Compute_Matched_No_B(X, AG)   is
  Input
    G = (N, E) - the local social network;
    X - a node from G;
    AG = (AN, AE) - a clustered anonymized social
         network of a social network G'=(N', E');
      N ⊆ N';  E ⊆ E';
  Output
    Matched_No_B(X, AG);

Part A
  $CL_1$ = ∅;
  For each cl ∈ AN
    If (X ∈ gen(cl)) then
      /*  using $A_1$, .., $A_q$ attributes) */
    $CL_1$ = {cl} ∪ $CL_1$;
Part B
  $CL_2$ = ∅;
  For each cl ∈ $CL_1$
    If (check_degree(X, cl) == true) then
      $CL_2$ = {cl} ∪ $CL_2$;
Part C
  $CL_3$ = ∅;
  For each cl ∈ $CL_2$
     /* Neighbors_cl - is a set of clusters that
        contain all clusters where neighbors of
        a node from cl are; this includes cl. */
    Neighbors_cl = ∅;
      If (|$IE_{cl}$| > 0) then
                 Neighbors_cl = {cl} ∪ Neighbors_cl;
      For each cl' ∈ AN
        If (|$IE_{cl,cl'}$| > 0) then
                     Neighbors_cl = {cl'} ∪ Neighbors_cl;
    For each neighbor Y of X
         /* Y can be a neighbor of a node from cl
         using degree and attribute's values
         ($A_1$, .., $A_q$ attributes only) */
      For each cl' ∈ Neighbors_cl
        If (check_degree(Y, cl') == true) and
          (Y ∈ gen(cl')) then
          $CL_3$ = {cl} ∪ $CL_3$;
           Break;
  Return the number of nodes that are contained in clusters
  from $CL_3$;
End Algorithm.
```

In parts B and C of the above algorithm in order to select only matching clusters based on possible degree of their nodes (function *check_degree*) we use the following property.

Property 5. Given an anonymized social network $\mathcal{AG}$, the degree of a node X that belongs to a cluster cl_j with $|cl_j| \geq 2$ is bounded by $[\mathit{min\_deg}(X), \mathit{max\_deg}(X)]$ where:

$$\mathit{min\_deg}(X) = \max\left(0, |IE_{cl_j}| - \frac{(|cl_j|-1)\cdot(|cl_j|-2)}{2}\right) + \sum_{\substack{i,j=1,v\\ i\neq j}} \max(0, |IE_{cl_i cl_j}| - (|cl_j|-1)\cdot|cl_i|),$$

$$\mathit{max\_deg}(X) = \min((|cl_j|-1), |IE_{cl_j}|) + \sum_{\substack{i,j=1\cdots v\\ i\neq j}} \min(|cl_i|, |IE_{cl_i cl_j}|).$$

Proof. The maximum possible degree for a node X within a cluster cl_j is the smaller value of the number of nodes in this cluster minus 1 and the number of internal edges, plus the sum of all maximum possible external edges connected to this node. Given a cluster cl_i which has external edges to cl_j, the number of maximum possible external edges connected to X is the smaller value of the number of nodes in cl_i and external degree $|IE_{cl_i cl_j}|$. To determine the minimum possible degree of a node, we take two steps. In the first step, we consider the structure within the cluster only. First, we try to assign the internal edges as many as possible to other nodes; the maximum number of edges we can assign to other nodes is

$$\frac{(|cl_j|-1)\cdot(|cl_j|-2)}{2}.$$

The number of internal edges minus the maximum number of edges that we can assign to other nodes is the number of edges we have to assign to X. However, the degree cannot be less than 0. In the second step, we also consider the external structure. Given a cluster cl_i, the maximum number of external edges we can assign to other nodes is

$$(|cl_j|-1)\cdot|cl_i|.$$

The number of external edges to cl_i minus the maximum number of edges that we can assign to other nodes is the number of edges we have to assign to X. However, the degree of a node cannot be less than 0. □

Note that for a cluster formed by a single node, the number of edges is already known since all inter-cluster edges are connected to the only node in the cluster, and it is equal to $\sum_{i=1\cdots v; i\neq j} |IE_{cl_i cl_j}|$. However, in a k-anonymized graph such clusters do not exist since the size of a cluster must be at least k.

In the *Compute_Matched_No_B* algorithm we consider a node X from the local graph that matches a cluster from the anonymized graph. Since the local graph is a subgraph of the original graph from which the anonymized graph was created, the degree of X in the local graph is less than or equal to the degree of X in the original graph. Therefore the min_deg(X) from Property 5 may not be a lower bound of X and we use value 0 instead in the function *check_degree*. On the other hand, max_deg(X) from Property 5 is used in this function.

Using the *Compute_Matched_No_B* algorithm we may slightly overestimate the number of matches. It is possible that a node will not match a cluster because of a mismatch between one of the nodes located at a distance 2 or more. One way we address this problem is to consider all possible graphs (also known as possible worlds[7]) that can be generated from $\mathcal{AG}$ and then for each such graph and the given local graph to compute the number of subgraphs with the same structure and profile as the local graphs that are found in the generated graph. This is the well-known subgraph isomorphism problem which was proven to be NP-complete.[24] While this overestimate may slightly change the values of trust scores, it is important to notice that the trust score will just increase. Thus, any trust score of 0 is fully reliable.

The algorithms for computing *matched_B*$(X, \mathcal{AG})$ and *matched_weighted_B*$(X, \mathcal{AG})$ are similar. For *matched_B*$(X, \mathcal{AG})$, for each cluster *cl* from CL_3 found by *Compute_matched_No_B*$(X, \mathcal{AG})$ we test if node X belongs to *gen*(cl) using the B attribute. The function returns the number of nodes that are contained only in the clusters that satisfy this test. For *matched_weighted_B*$(X, \mathcal{AG})$, in addition to this new test we also weigh each node based on the amount of generalization for the attribute B in the containing cluster.

The weight associated with a node will be always in the interval $(0, 1]$. The value of 1 is obtained when the value of the B attribute from the node is identical to the B attribute value from the matching cluster. In other words, this value was not generalized in that cluster. The data owner can define such weights on the value generalization hierarchy for the attribute B (and if such a hierarchy does not exist the data owner can create one for this purpose[35]). There are just two properties that these weights must follow. First, all the leaf values must have a weight of 1, and second, each parent node must have a lower weight than that of all its children. To simplify the selection of weights, in our experiments, we chose an automated approach to generate weights that satisfy these properties. With each node in the value generalization hierarchy, we associate the value 1/*no_contained_values*, where *no_contained_values* are all values that are descendants of that node.

The running time for part A in *Compute_matched_No_B*$(X, \mathcal{AG})$ is proportional to the number of clusters, $\lceil |N|/k \rceil$, in the anonymized graph $\mathcal{AG}$. For part B, the running time is the same. Thus the complexity of parts A and B is O($|N|$). For part C, in the worst case, the number of neighbors that are checked for a given node X is $|N| - 1$ and the number of clusters is $\lceil |N|/k \rceil$. The complexity of part C is O($|N|^2$) and the overall complexity of this algorithm is O($|N|^2$).

6. Trust Scores for k-Degree Anonymous Social Network

6.1. *Anonymization model*

The second anonymization model we use for our trust computation model for social networks is the k-degree anonymous social network model.[5] Its definition is below.

Definition 9 (k-degree anonymity). A graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is k-degree anonymous if for every node $X \in \mathcal{N}$ there exist atleast $k - 1$ other nodes that have the same degree as X.

An algorithm that creates a k-degree anonymous graph is also presented in Ref. 5. Given a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, this algorithm attempts to output a k-anonymous graph $\mathcal{G}' = (\mathcal{N}, \mathcal{E}')$ such that (a) the L_1-norm of the difference of their degree sequence is minimized and (b) the symmetric difference of their sets of edges is also minimized.

To be more specific, the algorithm consists of two main steps. For an input graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with degree sequence $\mathbf{d}$, and an integer k, starting from $\mathbf{d}$, it first constructs a new degree sequence $\mathbf{d}'$ such that the L_1-norm distance between $\mathbf{d}$ and $\mathbf{d}'$ is minimized. Second, it tries to construct a graph $\mathcal{G}' = (\mathcal{N}, \mathcal{E}')$ such that the difference between their sets of edges is minimized. If only the addition of edges is allowed, $\mathcal{G}'$ will be a supergraph of $\mathcal{G}$ and what the algorithm attempts to minimized is $|\mathcal{E}'\backslash\mathcal{E}| = |\mathcal{E}'| - |\mathcal{E}|$.

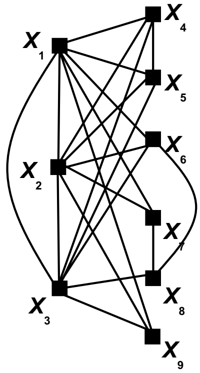
Since nodes in the social network graph considered in our paper are also associated with attribute values, we need to generalize these values before publishing the graph in order to protect the privacy of individual nodes. Here we use the groups induced by such an algorithm to generalize the attribute values. Specifically, for each group of size less than $2k$, we simply generalize the attribute values. If a group is of size $g > 2k$, it will be partitioned sequentially into $\lfloor g/k \rfloor - 1$ subgroups of size k and a last subgroup of size $g - k(\lfloor g/k \rfloor - 1)$. Subsequently, the attribute values in each subgroup described above are generalized accordingly.

6.2. *Intermediary trust scores computations*

In this section, we present a method to compute the intermediary trust scores when the k-degree anonymous model is used.

We start by continuing Example 1 from Sec. 5.2.

Example 1 — Cont. Assume that the data owner has the local social network $\mathcal{G}_1$ given in Fig. 5. Suppose that the data owner has access to the three-anonymous social network $\mathcal{AG}_3$ in Fig. 6. The data owner fully trusts the values of *age* and *sex* attributes, and wants to assess the trustworthiness of the attribute *city*. All other assumptions remain the same as in the example from Sec. 5.2. The graph in Fig. 6 is a three-degree anonymous network of the social network in Fig. 2 which is produced using Liu and Terzi's algorithm where only edge additions are allowed. Several edges



	Sex	Age	City
X_1	P	20-30	Detroit
X_2	P	20-30	Detroit
X_3	P	20-30	Detroit
X_4	P	24-35	US
X_5	P	24-35	US
X_6	P	24-35	US
X_7	F	33-45	US
X_8	F	33-45	US
X_9	F	33-45	US

Fig. 6. The three-anonymous social network $\mathcal{AG}_3$.

are added to the anonymized graph (see Fig. 6) in order to meet the three-degree anonymity. Also, the quasi identifier attribute values must be generalized in each group to the most specific common values.

In order to compute the trust scores, the first step is to match each node from the local graph to nodes from the anonymized graph. We perform two types of matches. First, we use the graph structure and the *sex* and *age* attributes values. Second, we also add the non-trusted attribute *city* to this match process. Algorithms for both procedures will be detailed later in this section. After the matches, we can directly compute intermediary absolute, and relative trust scores using Definitions 1 and 2. We define the weights as in the example from Sec. 5.2. Table 4 shows the intermediary trust score computations for $\mathcal{AG}_3$.

If we have more than one anonymized graph (note that these graphs can be obtained by different anonymization models), the combined procedure is similar to the one presented in Sec. 5.2.

We compute $matched\_No\_B(X, \mathcal{AG})$ for this model as shown in the below algorithm. The algorithms to compute the other two values, $matched\_B(X, \mathcal{AG})$ and $matched\_weighted\_B(X, \mathcal{AG})$ are similar.

Table 4. The intermediary trust scores for $\mathcal{AG}_3$.

	Nodes Matched Using City	Nodes Matched No City	*matched_B*	*matched_ No_B*	*matched_ weighted_B*	*ATS*	*RTS*	*WTS*
X_1	∅	X_1, X_2, X_3	0	3	0	0	0	0
X_2	X_1, X_2, X_3	X_1, X_2, X_3	3	3	3	1/3	1	1
X_3	X_4, X_5	X_1, X_2, X_3, X_4, X_5	2	5	1/2	2/9	2/5	1/10
X_4	X_4, X_5	X_1, X_2, X_3, X_4, X_5	2	5	1/2	2/9	2/5	1/10
X_5	X_4, X_5, X_6	X_4, X_5, X_6	3	3	3/4	1/3	1	1/4
X_9	X_7, X_8, X_9	X_7, X_8, X_9	3	3	3/4	1/3	1	1/4

```
Algorithm Compute_Matched_No_B_Liu(X, AG)   is
  Input
    G = (N, E) - the local social network
    AG = (AN, AE) - a $k$-degree anonymized social network of
    G'=(N', E') (we drop subscript $j$ for simplicity)
    E' $\subseteq$ AE (only edge additions are performed)
    N $\subseteq$ AN; E $\subseteq$ E'
  Output
    Matched_No_B(X, AG); // Using Liu Model

Part A
    MatchingNodes$_1$ = $\varnothing$;
    For each $X_i \in$ AN
      If $X_i$ matches the values from $X$ for
      ($A_1$, .., $A_q$ attributes)  then
      MatchingNodes$_1$ = $\{X_i\} \cup$ MatchingNodes$_1$;
Part B
    MatchingNodes$_2$ = $\varnothing$;
    For each $X_i \in$ MatchingNodes$_1$
      If degree($X$) <= degree($X_i$) then
      //only edge additions allowed
      //$X_i$ could potentially have the same number
      //of neighbors as $X$
        MatchingNodes$_2$ = $\{X_i\} \cup$ MatchingNodes$_2$;
Part C
    MatchingNodes$_3$ = $\varnothing$;
    For each $X_i \in$ MatchingNodes$_2$
      For each neighbor $Y$ of $X$
        If $Y$ can be a neighbor of $X_i$ using degree and
        attribute's values ($A_1$, .., $A_q$)then
        MatchingNodes$_3$ = $\{X_i\} \cup$ MatchingNodes$_3$;
    Return |MatchingNodes$_3$|
End Algorithm
```

The running time for part A in *Compute_Matched_No_B_Liu* $(X, \mathcal{AG})$ is proportional to the number of groups, $\lceil |N|/k \rceil$, in the anonymized graph $\mathcal{AG}$. For part B, the complexity in the worst case is the size of $\mathcal{AG}$. Thus the complexity of parts A and B is O($|N|$). For part C, in the worst case, the number of neighbors that are checked for a given node X_i is $|N| - 1$. Thus, the complexity of part C is O($|N|^2$) and the overall complexity is O($|N|^2$).

7. Partially Trusted Anonymized Social Networks

Until now, when computing *matched_B*(X, AG_j), we assumed that the anonymized social networks were fully trusted. This assumption can be relaxed by assigning a trust score to each anonymized social network node. If an anonymized social network is not fully trustworthy, the owner of the anonymized social network can assign a pre-computed trust score to each anonymized social network node, labeled as $X \cdot TS$. If a clustered-based anonymization model is adopted, e.g. the one we adopted in Sec. 5, a trust score, computed as the average trust score within each cluster, is assigned to each cluster. We label the average trust score assigned to each cluster as $cl \cdot TS$. We modify *Compute_Matched_No_B*(X, AG) as follows:

```
Algorithm Compute_Matched_No_B(X, AG)   is
  Input
    G = (N, E) - the local social network;
    X - a node from G;
    AG = (AN, AE) - a clustered anonymized social
         network of a social network G'=(N', E');
      N ⊆ N';  E ⊆ E';
  Output
    Matched_No_B(X, AG);

   $CL_1$ = ∅;
   For each cl ∈ AN
     If (X ∈ gen(cl)) then
       /*  using $A_1$, .., $A_q$ attributes) */
      $CL_1$ = {cl} ∪ $CL_1$;
   $CL_2$ = ∅;
   For each cl ∈ $CL_1$
     If (check_degree(X, cl) == true) then
       $CL_2$ = {cl} ∪ $CL_2$;
   $CL_3$ = ∅;
   For each cl ∈ $CL_2$
      /* Neighbors_cl - is a set of clusters that
         contain all clusters where neighbors of
         a node from cl are; this includes cl. */
     Neighbors_cl = ∅;
       If (|$IE_{cl}$| > 0) then
                  Neighbors_cl = {cl} ∪ Neighbors_cl;
       For each cl' ∈ AN
         If (|$IE_{cl,cl'}$| > 0) then
                      Neighbors_cl = {cl'} ∪ Neighbors_cl;
     For each neighbor Y of X
          /* Y can be a neighbor of a node from cl
          using degree and attribute's values
          ($A_1$, .., $A_q$ attributes only) */
          For each $cl' \in Neighbors\_cl$
             If ($check\_degree(Y, cl')$ == true) and
                ($Y \in gen(cl')$) then
                $CL_3 = \{cl\} \cup CL_3$;
                 Break;
   $NWN$ = 0;
    For each $cl \in CL_3$
      $NWN = NWN + |cl| \times cl$.TS
   Return $NWN$ as the number of weighted nodes that are
   contained in clusters from $CL_3$;
End Algorithm.
```

Algorithm Compute_Matched_No_B_Liu(X, AG) can be modified in a similar way.

8. Experimental Results

In this section, we evaluate the effectiveness and efficiency of our trust model and algorithms. The language used for the experiment implementation is JAVA. The experiments were performed on an Intel(R) Core2 2.66 GHz workstation with 4 GB memory, running Windows 7. We used two datasets in the experiments: the synthetic AB datasets (for Albert-Barabasi); and the Enron datasets, which is an email network from Enron Corporation.[36] For each dataset, we selected 5,000 and 10,000 nodes as our initial datasets. To generate the synthetic dataset, we used the Albert-Barabasi model[37] to generate a random graph that follows the power-law distribution. For the Enron datasets the numbers of edges are 67,283 (for 5,000 nodes) and 110,615 (for 10,000 nodes). For the AB datasets the numbers of edges are 97,503 and 197,248 (for 5,000 and 10,000 nodes, respectively).

To fit the need of our experiments, we extended the synthetic and real datasets by introducing the following attributes: *age*, *sex*, *marital status*, and *city*. We use a simple program to generate the attribute values for the synthetic and real social networks described above. To do this, we randomly select a single node and assign to this node values for all four attributes. Next, we run the *breath first search* (BFS) algorithm starting from this source node. Each time when a new node v is first discovered by BFS, we generate its *age* and *city* according to its parent u. For the *age* attribute, we define six possible intervals: [18, 27], [28, 37], [38, 47], [48, 57], [58, 67], and [68, 77]. We consider the probability that v is within the same *age* interval as its parent u to be 0.7. The probability that v is in any other interval used by our algorithm is 0.3/5. After the *age* interval of node v is decided, its actual *age* will be a randomly chosen integer within that interval. For *gender* attribute, a node is assigned the *Male* value with probability 0.5 and the *Female* value with probability 0.5. To generate the values of *marital status* attribute for a node v, if v is of *age* greater than or equal to 70, with probability 0.5, its *marital*

Table 5. Hierarchy structure of the city attribute.

City	East	MA	Boston
		FL	Tampa, Miami
	Midwest	MI	Detroit
		IL	Chicago
		IN	Lafayette, Indianapolis, Bloomington
		OH	Cincinnati, Columbus, Cleveland
		KY	Lexington, Louisville, Frankfort, Newport
	Mountain	CO	Aspen
	West	CA	Sacramento, Riverside
		WA	Seattle, Redmond

status will be *Widowed*, and with probability 0.5, its *marital status* will be randomly generated as one of one of the following values: *Divorced*, *Never-married*, *Separated*, *Married-civ-spouse*, *Married-spouse-absent*, *Married-AF-spouse*. If v is of *age* less than 70, then v's *marital status* will be generated as one of the values above with equal probability. For the *city* attribute, we consider 20 values as shown in Table 5. Similarly, the probability that v is in the same city as its parent u is 0.7. The probability that v is in any other 19 cities is 0.3/19.

For each experiment, from an original dataset (*Enron* or *AB*; each with size 5,000 or 10,000) we generate one *local dataset* and one or more *remote datasets*. The local dataset is generated by taking 20% of the synthetic dataset/real dataset and modifying some of the records in them to create fake/incorrect values. In our experiments, the *city* attribute is the only attribute that contains fake values. In other words, it is the non-trusted attribute. The distance between two leaf nodes is defined as the height of their lowest common ancestor. For example, the distance between *Tampa* and *Miami* is 1, since their lowest common ancestor is *FL* and its height is 1. For the same reason, the distance between *Boston* and *Tampa* is 2 and distance between *Miami* and *Detroit* is 3. To generate the incorrect values, we take a portion (referred as p% in Table 6) of nodes in the local dataset and modify them according to a parameter (m) that controls distances. Among the candidates that share same distance between the original values, we randomly pick one.

For remote datasets, we first take the same portion (20%) of the original dataset as the local dataset without injecting any fake values. Then, we randomly pick remaining nodes in the original dataset until the size of the remote datasets reaches 80% of the original ones. Using this approach, the remote datasets always contain the corresponding local dataset nodes and structure.

Table 6. Experiment parameters.

p	10%, **20%**, 30%, 40%, 50%
m	1, 2, **3**
k	3, **5**, 10, 15, 20
s	**5000**, 10,000

Table 6 lists the parameters used in the experiments. p controls the percentage of fake values injected into the local dataset. m is the magnitude parameter of the fake values. For instance, when $m = 3$, all distances between fake values and their original ones are 3. When a *city* attribute value does not have a sibling node with $m = 1$ or 2 (e.g. *Boston* and *Aspen*), we will not modify its value and skip that node. k is the k-anonymity parameter for generating anonymization graphs. s is the number of nodes in the original dataset. In all our experiments when a parameter value is not specified, the one denoted in boldface in Table 6 is used.

In our experiments we use two quantities to measure the effectiveness of our trust computation algorithms: (1) the ratio of average score of unmodified nodes (*real*) to the average score of modified nodes (*fake*), denoted as *Score Ratio* in the figure; and (2) the *Recall* that is the percentage of modified nodes that have been classified as "fake". To classify the local nodes, we simply rank the nodes with respect to their *ATS*, *RTS*, and *WTS* and choose either p% or $1.5p$% nodes with the lowest trust scores as possible fake nodes. The data owner will then perform an additional investigation regarding these nodes to determine which of the determined nodes have fake values for the target attribute. For example, if $p = 10\%$ and size of the local dataset is 1000, we will classify 100 or 150 nodes with the lowest trust scores as fake nodes. It is worth noting that when no trust score is used and we do a random sampling to choose those 100 or 150 nodes, only p% (10 or 15 if $p = 10\%$) of nodes have fake values.

Due to space limitation, all figures we reported below are based on *Enron* dataset with size 5,000. We observe almost identical trends in the experimental results on *Enron* dataset with size 10,000 and on synthetic *AB* datasets with sizes 5,000 and 10,000.

In the experiments reported in Fig. 7, one k-anonymous clustered social network (see Sec. 5) is used with the local graph to compute the trust scores. We observe the effects of changing p on the score ratio and the recall. We notice that the *WTS* score ratio is the highest among the three score ratios regardless of the p value. As for the recall, we observe that all three scores are effective and, among them, *WTS* performs the best. Both *ATS* and *RTS* have more than 70% recall and *WTS* has

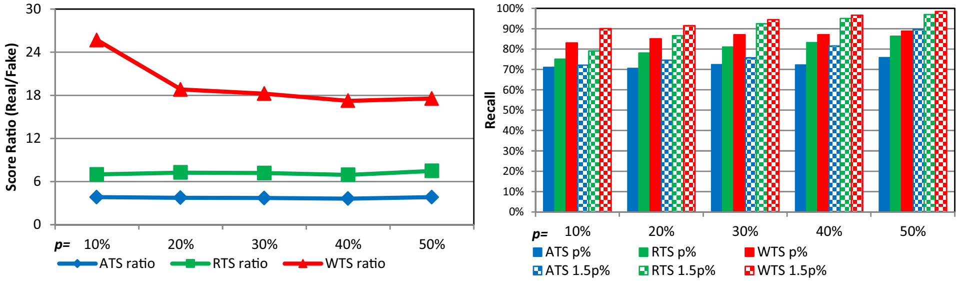


Fig. 7. One k-anonymous clustered social network, change p, $m = 3$, $s = 5{,}000$, Enron dataset.

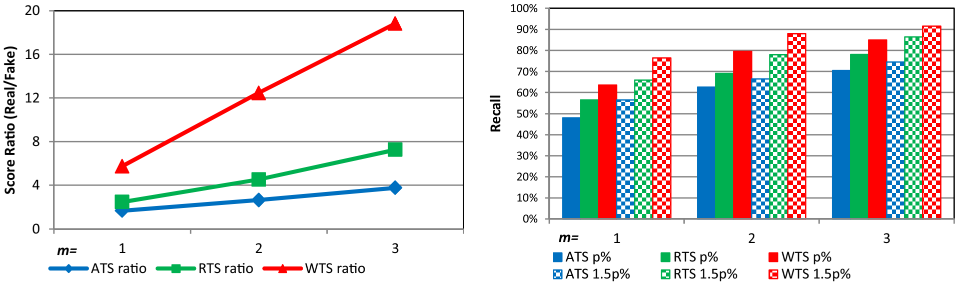


Fig. 8. One k-anonymous clustered social network, change m, $p = 20\%$, $s = 5{,}000$, Enron dataset.

more than 80% recall when $p = 10\%$ and it increases when p grows. When $p = 10\%$, our approach can increase the recall by more than seven times when ATS or RTS are used and more than eight times when WTS is used, which demonstrates that our methods are very effective.

In the experiments reported in Fig. 8, one k-anonymous clustered social network is used and p is set to 20%. We can see the effects of changing m on the score ratio and the recall. When m increases, all the score ratios increase, which is as expected because the larger the magnitude of error we inject into a node, the more easily this fake node will be detected. For the recall, it is obvious that recall increases with m and that WTS performs the best among those three scores. We can also observe that even for $m = 1$, the recall values are over 40% (ATS), 50% (RTS) and 60% (WTS) (the baseline recall is 20%).

In the experiments reported in Fig. 9, we test how different k values impact the trust score computation results. One k-anonymous clustered social network with different k values is used. Both score ratio and recall decrease when k increases. This is due to the fact that larger k values increase information loss. We can also observe that WTS is more affected by k values. This is because WTS takes attribute weights into consideration. When the data is more generalized, it is more difficult for WTS to distinguish fake values from real ones, since the weight associated with each generalized values is very small.

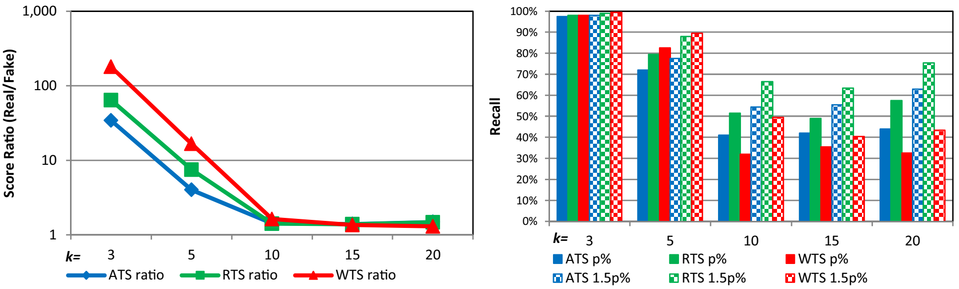


Fig. 9. One k-anonymous clustered social network, change k, $p = 20\%$, $m = 3$, $s = 5{,}000$, Enron dataset.

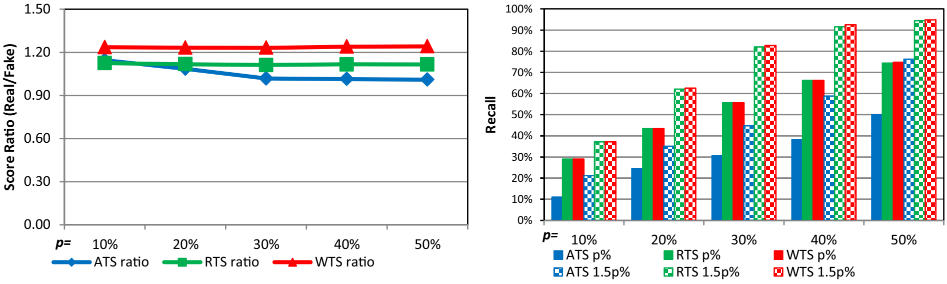


Fig. 10. One k-degree anonymous social network, change p, $m = 3$, $s = 5{,}000$, Enron dataset.

In the experiments reported in Fig. 10, one k-degree anonymous social network (see Sec. 6) is used with the local graph to compute the trust scores for local nodes. Among those three different score measures, *WTS* score ratio and recall are always the highest and *RTS* is close to *WTS*. *ATS* performs a lot worse in this scenario. In most cases, its recall is close to the value of p%, which means that it is not better than a random sampling. On the other hand, *RTS* and *WTS* are able to increase the recall by more than 20% in all cases. It is important to notice that the score ratios and recalls are much lower than those in the experiments that use k-anonymous clustered social networks (Figs. 7–9). This is due to the anonymization procedure; the k-degree anonymous social network approach does not consider the attribute values in the graph generation process and this leads to more coarse generalization.

In the experiments reported in Fig. 11, one k-degree anonymous social network is used and p is set to 20%. We can see the effects of changing m on the score ratio and the recall. When m increases, all the score ratios increases, which is as expected because the larger the magnitude of error we inject into a node, the more easily this fake node will be detected. For the recall, it is obvious that recall increases with m and that *WTS* performs the best among those three scores when $m = 2, 3$. We can also observe that for $m = 3$, the recall values are over 40% for *RTS* and *WTS* (the baseline recall is 20%).

In the experiments reported in Fig. 12, we test how different k values impact the trust score computation results for k-degree anonymous social networks. One

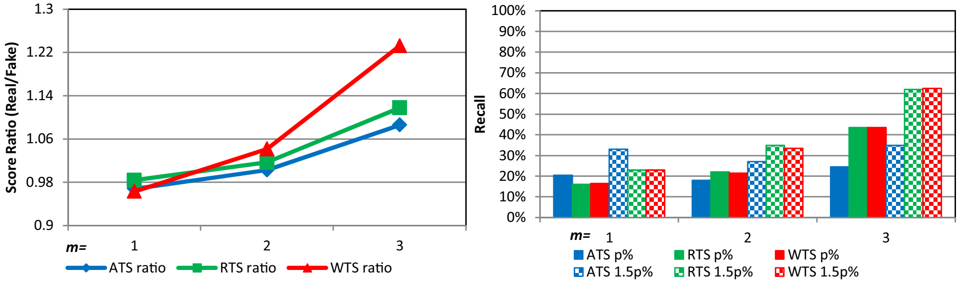


Fig. 11. One k-degree anonymous social network, change m, $p = 20\%$, $s = 5{,}000$, Enron dataset.

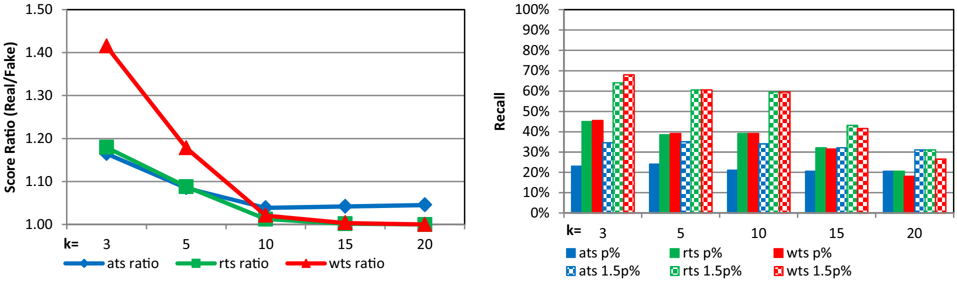


Fig. 12. One k-degree anonymous social network, change k, $p = 20\%$, $m = 3$, $s = 5{,}000$, Enron dataset.

k-degree anonymous social network with different k values is used. Both score ratio and recall decrease when k increases. This is due to the fact that larger k values increase information loss. We can also observe that WTS is more affected by k values. This is because WTS takes attribute weights into consideration. When the data is more generalized, it is more difficult for WTS to distinguish fake values from real ones, since the weight associated with each generalized values is very small.

In the experiments reported in Fig. 13, one k-degree anonymous social network is used together with one k-anonymous clustered social networks with different k values. We can observe that both score ratio and recall increase as k decreases. As expected, with smaller k, more information is preserved in the anonymized graphs. We can also observe that WTS does not perform as well as RTS. This is due to the same reason as for the results from the experiments reported in Fig. 9, that is, that WTS is more sensitive to the value of k.

In the experiments reported in Fig. 14, we test the running time of our trustworthiness computation algorithms for both privacy models (k-anonymous clustered social networks and k-degree anonymous social networks), different values of k, and dataset sizes with 5,000 (1,000 local nodes and 4,000 remote nodes) and 10,000 nodes (2,000 local nodes and 8,000 remote nodes). We observe that the running time first increases with k and this is because when k grows, it introduces more

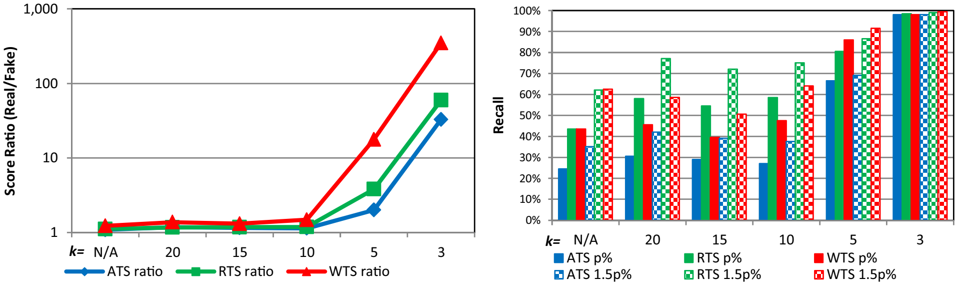


Fig. 13. One k-degree anonymous social network with a k-anonymous clustered social network, $s = 5{,}000$, Enron dataset.

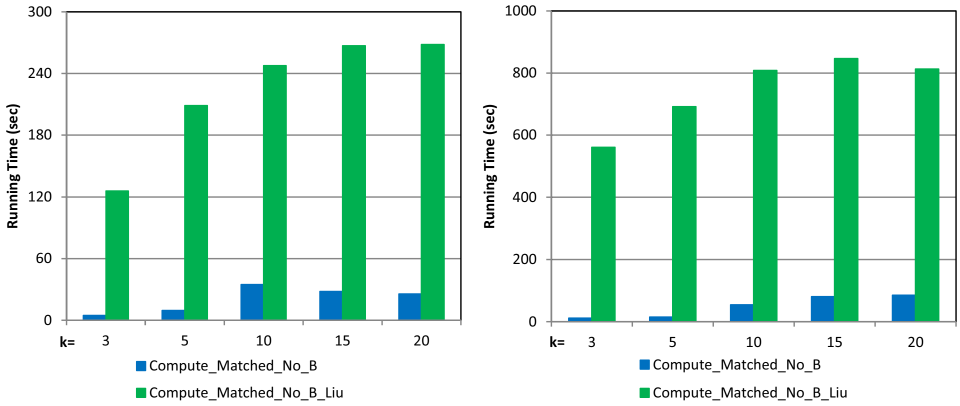


Fig. 14. Running time.

neighbors. However, when k reaches a threshold, the running time becomes stable and it starts to decrease slightly since the number of neighbors is no longer increasing. We also observe that the trustworthiness computation algorithm for k-anonymous clustered social network runs much faster than the trustworthiness computation algorithm for k-degree anonymous social network. This is due to the fact that k-anonymous clustered social network contains less nodes and does not add more edges to the network. It is worth mentioning that both algorithms are much more efficient than the anonymization algorithms used to create anonymized social networks. For example, when $k = 20$, the k-degree anonymization algorithm needs 31,883 s ($\sim$8.85 h) and the clustered-based anonymization algorithm needs 5,197 s ($\sim$1.44 h) to anonymize a 8,000 nodes social network while it only takes the *Compute_Matched_No_B* algorithm less than 100 s and the *Compute_Matched_No_B_Liu* algorithm 900 s to compute the trust scores.

9. Conclusions and Future Work

In this paper, we have shown the importance of taking into account privacy when computing trustworthiness of social network data in applications such as deciding insurance rate. We have proposed three trust score computation models (absolute, relative, and weighted) and algorithms for computing them based on two different anonymization models. An extensive experimental evaluation on both real and synthetic data has demonstrated the effectiveness and efficiency of our approach.

In future work, we plan to investigate how to extend our techniques to multiple non-trusted attributes and unknown non-trusted attribute. We also plan to develop new approaches similar to attributes prediction[29] to see whether attribute values of a local node match our prediction. However, our goal is to create prediction models based on remote anonymized datasets and use those models to determine the trustworthiness of local datasets, which is more challenging.

References

1. L. Freeman, *The Development of Social Network Analysis* (Empirical Press, Vancouver, 2006).
2. Q. Yang and X. Wu, 10 challenging problems in data mining research, *Int. J. Inform. Technol. Decision Making* **5**(4) (2006) 597–604.
3. D. MacMillan, The big trend in social network sites, BussinesWeek, September 9, 2010.
4. A. Campan and T. M. Truta, A clustering approach for data and structural anonymity in social networks, in *PinKDD*'08 (2008).
5. K. Liu and E. Terzi, Towards identity anonymization on graphs, in *SIGMOD*'08 (2008), pp. 93–106.
6. J. Cheng, A. Fu and J. Liu, K-isomorphism: Privacy preserving network publication against structural attacks, in *SIGMOD*'10 (2010).
7. M. Hay, G. Miklau, D. Jensen, D. Towsley and P. Weis, Resisting structural reidentification in anonymized social networks, in *Proc. VLDB* (2008), pp. 102–114.
8. C. H. Tai, D. N. Yang, P. S. Yu and M. S. Chen, Structural diversity for privacy in publishing social networks, in *Proc. SDM* (2011).
9. W. Wu., Y. Xiao, W. Wang, Z. He and Z. Wang, K-symmetry model for identity anonymization in social networks, in *Proc. EDBT* (2010).
10. X. Ying and X. Wu, Randomizing social networks: A spectrum preserving approach, in *Proc. SDM* (2008).
11. B. Zhou and J. Pei, Preserving privacy in social networks against neighborhood attacks, in *Proc. ICDE* (2008).
12. J. Gehrke, E. Lui and R. Pass, Towards privacy for social networks: A zero-knowledge based definition of privacy, in *Proc. Theory of Cryptography Conference* (2011), pp. 432–449.
13. C. Dwork, F. McSherry, K. Nissim and A. Smith, Calibrating noise to sensitivity in private data analysis, in *Proc. Theory of Cryptography* (2006), pp. 265–284.
14. L. Backstrom, C. Dwork and J. M. Kleinberg, Wherefore art thou r3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography, in *Proc. of the WWW* (2007).
15. A. Narayanan and V. Shmatikov, Robust de-anonymization of large sparse datasets, in *Proc. IEEE Symp. Security and Privacy* (2008), pp. 111–125.
16. C. N. Ziegler and G. Lausen, Propagation models for trust and distrust in social networks, *Inform. Syst. Frontiers* **4**(5) (2005) 337–358.
17. R. Levien and A. Aiken, Attack-resistant trust metrics for public key certification, in *Proc. USENIX Security Symposium* (1998), pp. 229–242.
18. L. Mui, M. Mohtashemi and A. Halberstadt, A computational model of trust and reputation, in *Proc. Hawaii International Conference on System Science (HICSS)* (2002).
19. L. Katz, A new status index derived from sociometric analysis, *Psychometrika* **18** (1953) 39–43.
20. L. Page, S. Brin, R. Motwani and T. Winograd, The PageRank citation ranking: Bringing order to the web, Stanford Digital Library Technologies Project, Technical Report, 1998.
21. J. M. Kleinberg, Authoritative sources in a hyper-linked environment, *J. ACM* **6**(5) (1999) 604–632.
22. Honestly, www.honestly.com. 2010.
23. Duedil, www.duedil.com (2011).

24. S. A. Cook, The complexity of theorem-proving procedures, in *Proc. ACM Symp. Theory of Computing* (1971), pp. 151–158.
25. J. R. Ullman, An algorithm for subgraph isomorphism, *J. ACM* **23**(1) (1976) 31–42.
26. L. P. Cordella, P. Foggia, C. Sansone and M. Vento, A (sub)graph isomorphism algorithm for matching large graphs, *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(10) (2004) 1367–1372.
27. M. Gori, M. Maggini and L. Sarti, Exact and approximate graph matching using random walks, *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(7) (2005) 1100–1111.
28. P. Pedarsani and M. Grossglauser, On the privacy of anonymized networks, in *Proc. KDD* (2011), pp. 1235–1243.
29. A. Mislove, B. Viswanath, K. P. Gummadi and P. Druschel, You are who you know: Inferring user profiles in social networks, in *Proc. ACM Conf. Web Search and Data Mining* (2010).
30. M. Kim and J. Leskovec, Modeling social networks with node attributes using the multiplicative attribute graph model (2011), arXiv:1106.5053v1.
31. 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. WWW* (2009).
32. R. Heatherly and M. Kantarcioglu, Automatic sanitization of social network data to prevent inference attacks, in *Proc. WWW* (2011), pp. 55–56.
33. P. Samarati, Protecting respondents identities in microdata release, *IEEE Trans. Knowledge Data Eng.* **13**(6) (2001) 1010–1027.
34. L. Sweeney, K-anonymity: A model for protecting privacy, *Int. J. Uncertainty, Fuzziness, Knowledge-Based Syst.* **10**(5) (2002) 557–570.
35. A. Campan, N. Cooper and T. M. Truta, On-the-fly generalization hierarchies for numerical attributes revisited, in *Secure Data Management Workshop* (2011), pp. 18–32.
36. ENRON Dataset, Available at http://snap.stanford.edu/data.
37. A.-L. Barabasi and R. Albert, Emergence of scaling in random networks, *Science* **286** (1999) 509.