Charu C. Aggarwal, Yao Li, Philip S. Yu
IBM T J Watson Research Center
and
University of Illinois at Chicago

On the Hardness of Graph Anonymization

ICDM Conference, 2011

Introduction

- **Graph Anonymization:** How to release a network, while preserving the identities of the nodes?

- Problem arises from the context of network data sharing in social scenarios.

Challenges of De-Identification

- A straightforward method for de-identification simply strips the identification information from the nodes before release.

- Such an approach is not very effective, because the de-identified network can be matched with the original network, with the use of even partial information.

  – *L. Backstrom, C. Dwork, J. Kleinberg. Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography. WWW Conference, 2007.*

Existing Methods

- A natural approach for privacy-preservation is to perturb the structure of the network in order to prevent re-identification.

- Existing methods use techniques such as duplicating, removing, switching or grouping vertices or edges in input graphs.
  - Some of the techniques rely on completely random edge additions and deletions (Hay et al).
  - Others use more carefully designed anonymization methods (Liu and Terzi).

Existing Attack Methods

- A number of techniques have been proposed in the literature in order to attack anonymized social networks.
  - **Degree Based Attacks:** The signature of the ith order is represented by a set $Q_i(x)$. The set $Q_0(x)$ is the label of node x, the set $Q_1(x)$ is the degree of node x, the set $Q_2(x)$ is the multi-set of each neighbor's degree, and so on. This provides a unique node signature.
  - **Subgraph Attacks:** We assume that particular subgraph structures around a target node are known a-priori.
  - **Distance Attacks:** We assume that distances to a predefined set of nodes are known a-priori.
- Some techniques have been proposed to address such attacks.

Our Contributions

- Currently existing anonymization techniques may still not do the job.

- Large graphs may encode a fundamentally large amount of information in their co-variance structure

  – Inherently hard to anonymize

- Propose a systematic method which uses the underlying graph co-variance structure for privacy attacks.

Core Ideas

- The re-identification algorithm uses the aggregate *covariance behavior* of the network linkage structure.

- This attack measure is far more robust to edge perturbations as compared to typical utility measures such as the distances between nodes.
  - Our attack measure is based on *robust statistical quantifications which depend upon the aggregate network structure*, whereas typical utility measures such as distances are highly sensitive to the behavior of a few edges.
  - Utility is degraded much faster than privacy is achieved.

Notations and Definitions

- The graph G is denote by (V, E), where V is the set of vertices, and E is the set of edges.

- Assume that the edges in the graph are directed, though a similar analysis also applies to the case of undirected networks.

- The total number of vertices in the network is denoted by $|V| = N$.

- Real-world networks are typically *massive* and *sparse*.

Linkage Covariance

- For a given node i, let $\widehat{X}^i$ represent the random 0-1 variable, which takes on the value 1, if node i is linked by a directed edge to any particular (potentially adjacent) node and 0 otherwise.

- We have instantiations of this random variable for all possible (potentially) adjacent nodes j, and the corresponding instantiation is denoted by x_{ij}.

- The value of x_{ij} is 1, if a directed edge does indeed exist from node i to node j.

- The linkage covariance $LinkCov(p, q)$ between nodes p and q is equal to the covariance between the random variables $\widehat{X}^p$ and $\widehat{X}^q$.

Robustness of Linkage Covariance

- Linkage covariance is robust to edge additions and deletions for *massive* and *sparse* graphs.

- *Let L' be the estimated value of the link covariance between nodes p and q (with m_{pq} common neighbors) after the addition of edges with probability f_a. Then, we have:*

$$E[L'] = LinkCov(p, q) - 2 \cdot m_{pq} \cdot f_a / N$$

- *For deletion probability f_d, the expected value of the estimated link covariance L' is related to the true link covariance $LinkCov(p, q)$ as follows:*

$$E[L'] = LinkCov(p, q) \cdot (1 - 2 \cdot f_d)$$

Characteristic Vector

- Since the link covariances for a given node do not change very easily, they can be used to define a *signature or characteristic vector* for that node.

- The characteristic vectors for the nodes in the background knowledge graph and the de-identified graph can be matched with one another in order to create a re-identification attack.

- There are several ways of defining this signature or characteristic vector.

Possibilities

- When the mapping between the two graphs are completely unknown, we can create a vector of link covariances, which are sorted in decreasing values.

- When the mapping between the two graphs are approximately known for **all** nodes, we can define vector corresponding to the sort order of nodes in the two graphs:
  - This provides more accurate results, when we match the signatures between the two graphs.

- In some cases, an approximate mapping is known for some of the nodes, but not others.
  - We use a sort order on the nodes for which the mapping is known, and use a sort order on the magnitudes for others.

Matching Nodes in Graphs

- Determine an ordered set S' in the de-identified and perturbed graph G', so that the dot product of its characteristic vector with the characteristic vector of the ordered (identified) set S from the base (publicly available) graph G is maximized.

- The problem is hard to solve even in the case when the edges are not perturbed, because it is a version of the matching problem in graphs.

  – This problem is NP-hard.

- The characteristic vector provides an approximate way to match nodes with each other.

Bipartite Transformation

- We will transform the problem to a bipartite graph matching problem.

- Let $G = (V, E)$ and $G' = (V', E')$ be the original and de-identified graphs respectively.

- It is assumed that we have local information about a subset S of nodes of the original graph G.

- We create a bipartite graph $H = (V \cup V', D)$, where one partition of the bipartite graph contains nodes corresponding to V of G, whereas the other partition contains nodes corresponding to V' of G'.

Bipartite Transformation

- The edge set D contains an edge from each node of $S \subseteq V$ to every node of V'.

- The weight of an edge between $i \in V$ and $j \in V'$ is (initially) defined as the dot product of the normalized characteristic vectors for nodes i and j with respect to the entire vertex sets in the two graphs.

  - In the event that the graphs do not have the same number of vertices, we append zeros at the end of the smaller of the two characteristic vectors, so that a dot product can be performed.

Optimization

- For a given set $S \subseteq V$, a *bipartite matching* is defined as a set of node-disjoint edges, such that there is one edge from a node in S to a node in V'.

- Determine the maximum weight one-to-one matching from the subset of nodes S to the nodes in V' determines a matching of the nodes which maximizes the similarity between the corresponding nodes.

Optimization (Contd.)

- The first matching provides a correspondence of nodes in S and S'.
- This can be used to further refine the weights of the edges, by creating new sets of partially ranked characteristic vectors.
- Repeat process of matching and weight re-adjustment to convergence.

Experimental Results

- Tested the tradeoff between privacy and utility.
  - **(1) Distance perturbation (Utility Measure):** We sample k pairs of nodes, and compute the number of node pairs for which the change in distances between node pairs from the original to the perturbed graph is greater than one standard-deviation of the original distances between the node pairs.
  - **(3) Node Re-identification Rate (Privacy Measure):** Percentage of accurately identified nodes.

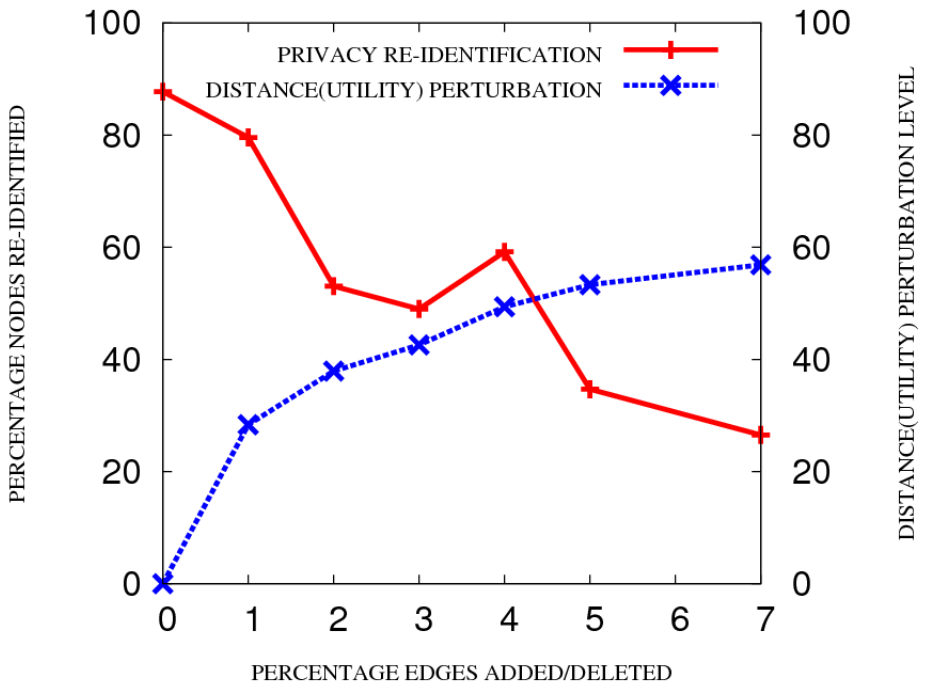
Data Sets

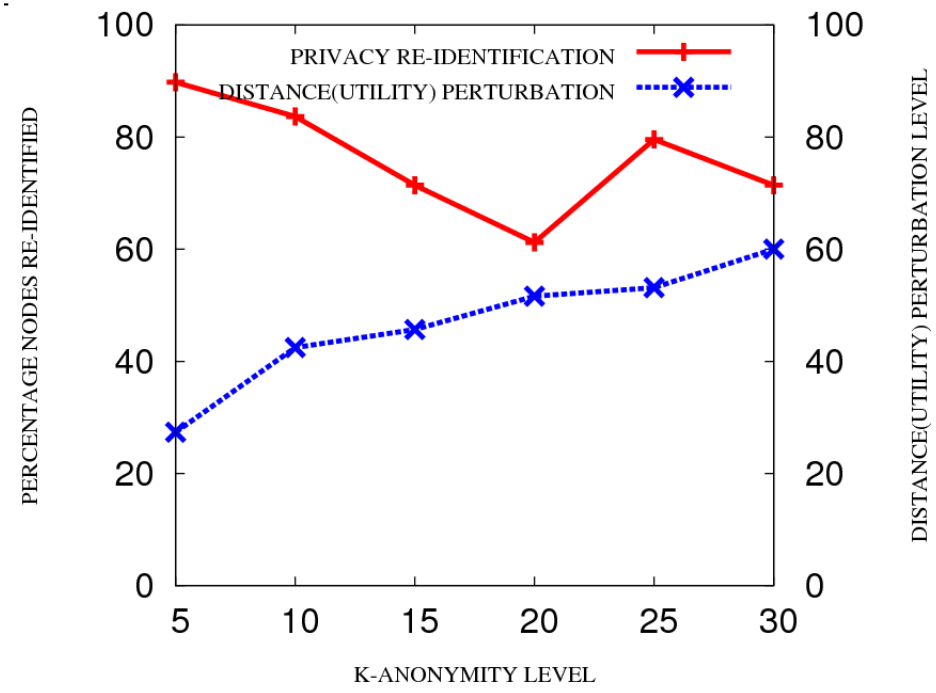
- **(1) Power Grid Graph:** Each vertex stands for a generator/transformer and substation and the edge between pairs of them represent the power line in the network.

- **(2) Co-author Graph:** This data set is a collection of bibliographies of scientific literature in computer science from various sources, covering most aspects of computer science. An edge will exist between a pair of authors who co-authored a paper.

- **(3) DIP (Database of Interacting Proteins):** The DIP database records experimentally determined interactions between proteins.

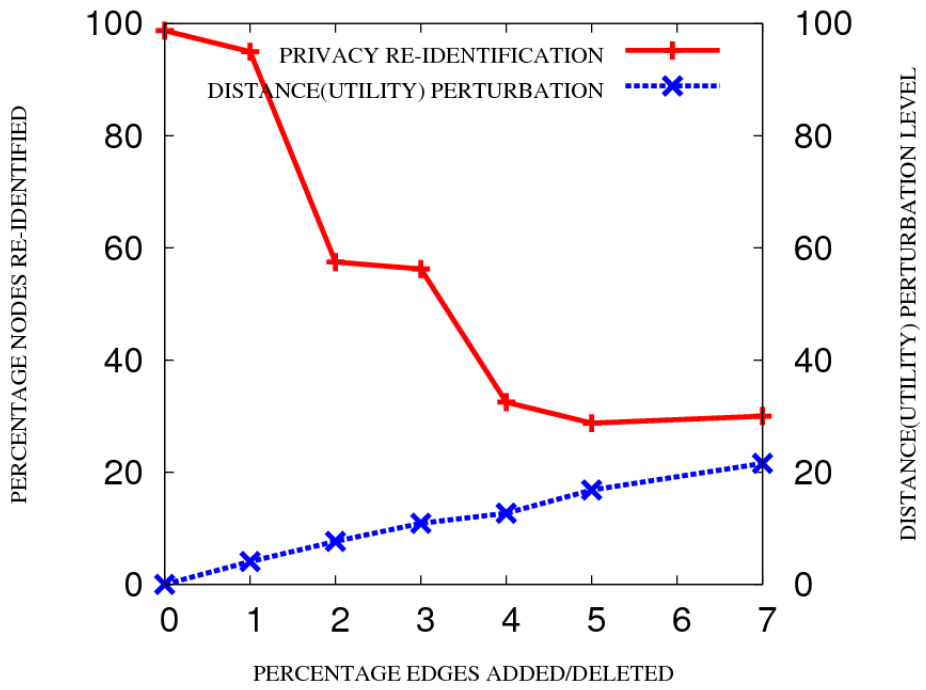
Distance (Utility) vs. Re-Identification Rate

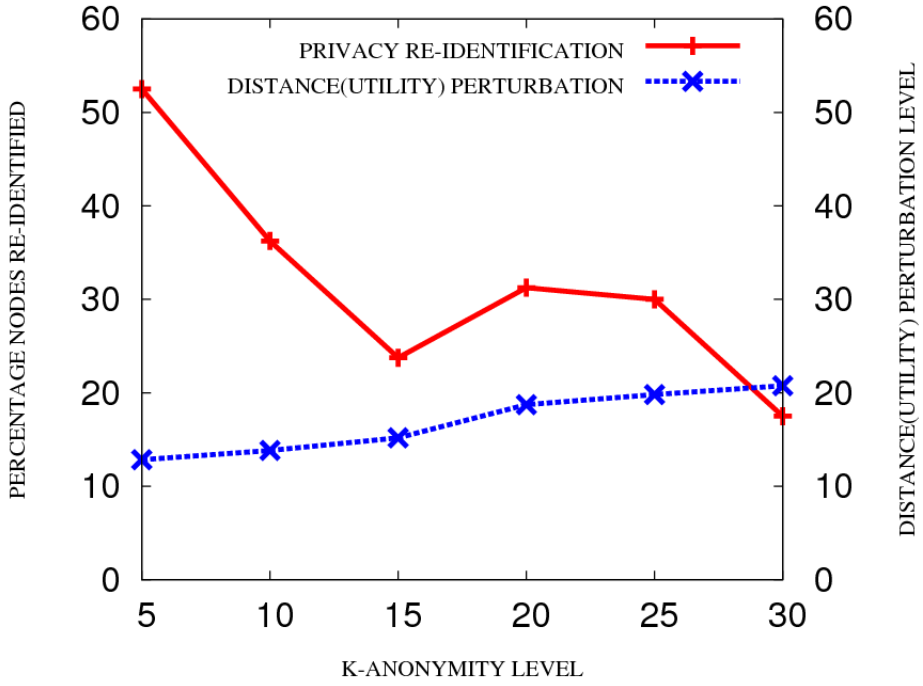




- Power Graph

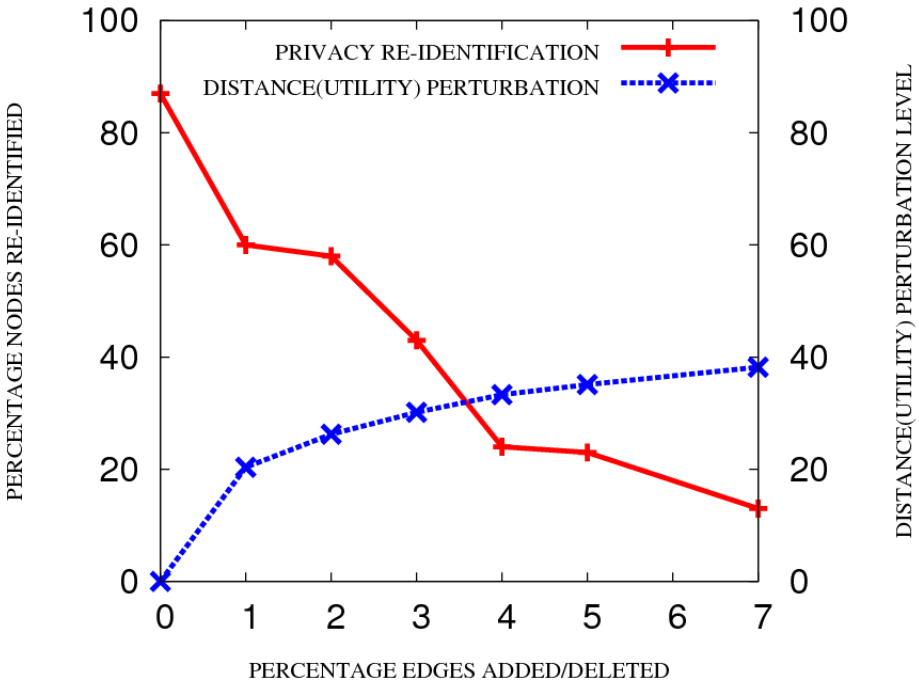
Distance (Utility) vs. Re-Identification Rate

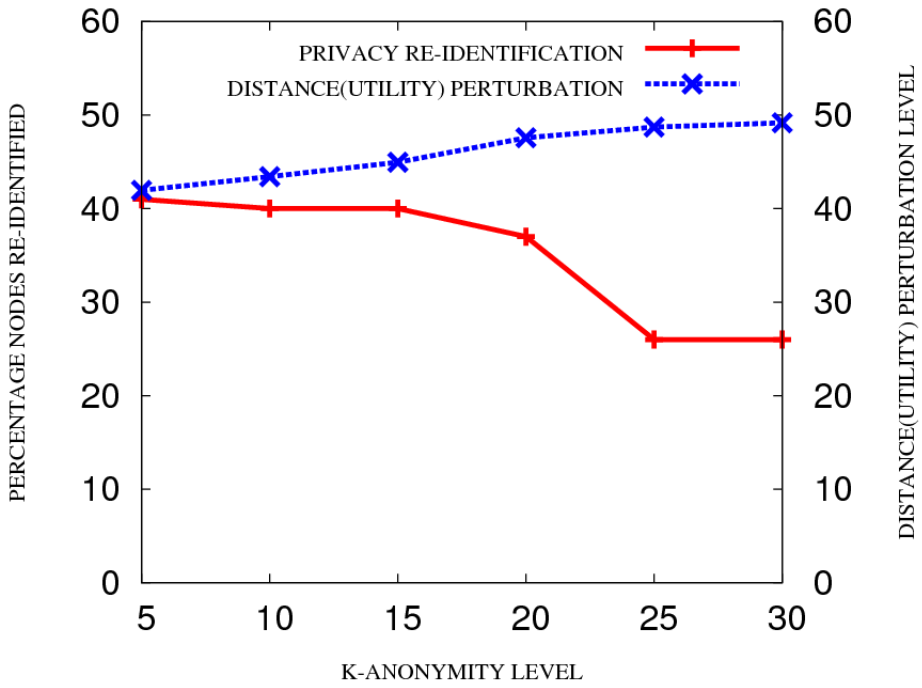




- Co-Author Graph

Distance (Utility) vs. Re-Identification Rate





- DIP Graph

Conclusions and Summary

- Theoretical results which show the hardness of graph anonymization.
- Present an algorithm for re-identification attack on the basis of these results.
- Presented experimental results illustrating the effectiveness of the approach.