Resisting Structural Re-identification in Anonymized Social Networks

Michael Hay, Gerome Miklau, David Jensen, Don Towsley, Philipp Weis
Department of Computer Science
University of Massachusetts Amherst
{mhay,miklau,jensen,towsley,pweis}@cs.umass.edu

ABSTRACT

We identify privacy risks associated with releasing network data sets and provide an algorithm that mitigates those risks. A network consists of entities connected by links representing relations such as friendship, communication, or shared activity. Maintaining privacy when publishing networked data is uniquely challenging because an individual's network context can be used to identify them even if other identifying information is removed. In this paper, we quantify the privacy risks associated with three classes of attacks on the privacy of individuals in networks, based on the knowledge used by the adversary. We show that the risks of these attacks vary greatly based on network structure and size. We propose a novel approach to anonymizing network data that models aggregate network structure and then allows samples to be drawn from that model. The approach guarantees anonymity for network entities while preserving the ability to estimate a wide variety of network measures with relatively little bias.

1. INTRODUCTION

A network data set is a graph representing a set of entities and the connections between them. Network data can describe a variety of domains: a *social network* describes individuals connected by personal relationships; an *information network* might describe a set of articles connected by citations; a *communication network* might describe Internet hosts related by traffic flows. As our ability to collect network data has increased, so too has the importance of analyzing these networks. Networks are analyzed in many ways: to study disease transmission, to measure the influence of a publication, and to evaluate the network's resiliency to faults and attacks. Such analyses inform our understanding of network structure and function.

However, many networks contain highly sensitive data. For example, Potterat et al. [21] published a social network which shows a set of individuals related by sexual contacts and shared drug injections. While society knows more about how HIV spreads because this network was published and analyzed, researchers had to weigh that benefit against possible losses of privacy to the individuals involved without clear knowledge of potential attacks. Other kinds of networks, such as communication networks are also considered sensitive. The sensitivity of the data often prevents the data owner from publishing it. For example, to our knowledge, the sole publicly available network of email communication was published only because of government litigation [7].

The objective of the data owner is to publish the data in such a way that permits useful analysis yet avoids disclosing sensitive information. Because network analysis can be performed in the absence of entity identifiers (such as name or social security number), the data owner first replaces identifying attributes with synthetic identifiers. We refer to this procedure as *naive anonymization*. It is a common practice in many domains, and it is often implemented by simply encrypting identifiers. Presumably, it protects sensitive information because it breaks the association between the sensitive data and real-world individuals.

However, naive anonymization may be insufficient. A distinctive threat in network data is that an entity's connections (i.e., the network structure around it) can be distinguishing, and may be used to re-identify an otherwise anonymous individual. In this paper, we investigate the threat of structural re-identification in anonymized networks. We consider how a malicious individual (the *adversary*) might learn about the network structure and then attempt to re-identify entities in the anonymized network. We formally model adversary capabilities, demonstrate successful attacks on real networks, and propose an improved anonymization technique.

Most existing work on privacy in data publishing has focused on tabular data, where each record represents a separate entity, and an individual may be re-identified by matching the individual's publicly known attributes with the attributes of the anonymized table. Anonymization techniques for tabular data do not apply to networked data because they fail to account for the interconnectedness of the entities (i.e., they destroy the network structure). It is not well-understood how publishing a network threatens privacy; initial investigations, including anonymization algorithms, are just emerging [4, 14, 17, 24, 26, 29, 30, 31].

Formally, we model a network as an undirected graph $G = (V, E)$. The naive anonymization of G is an isomorphic graph, $G_a = (V_a, E_a)$, defined by a random bijection $f : V \rightarrow V_a$. For example, Figure 1 shows a small network along with its naive anonymization. The *anonymization mapping* f, also shown, is a random, secret mapping.

Naive anonymization prevents re-identification when the adversary has no information about individuals in the original graph. Formally stated, an individual $x \in V$, called the *target*, has a *candidate set*, denoted cand(x), which consists of the nodes of G_a that could feasibly correspond to x. To assess the risk of re-identification, we assume each element of the candidate set is equally likely and use the size of the candidate set as a measure of resis-

PVLDB '08, August 23-28, 2008, Auckland, New Zealand

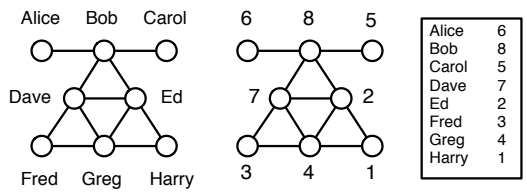


Figure 1: A social network (G), the naive anonymization (G_a), and the anonymization mapping (f).

tance to re-identification. Since f is random, in the absence of other information, any node in G_a could correspond to the target node x. Thus, given an uninformed adversary, each individual has the same risk of re-identification, specifically cand$(x) = V_a$ for each target individual x.

However, in practice the adversary may have access to external information about the entities in the graph and their relationships. This information may be available through a public source beyond the control of the data owner, or may be obtained by the adversary's malicious actions. For example, for the graph in Figure 1, the adversary might know that *"Bob has three or more neighbors,"* or that *"Greg is connected to at least two nodes, each with degree 2."* Such information allows the adversary to reduce the set of candidates in the anonymized graph for each of the targeted individuals. For example, the first statement allows the adversary to partially re-identify Bob: cand$(Bob) = \{2, 4, 7, 8\}$. The second statement re-identifies Greg: cand$(Greg) = \{4\}$.

Although an adversary may also have information about the attributes of nodes, the focus of this paper is *structural* re-identification, where the adversary's information is about graph structure. Re-identification with attribute knowledge has been well-studied, as have techniques for resisting it [18, 19, 25]. More importantly, many network analyses are concerned exclusively with structural properties of the graph, therefore safely publishing an unlabeled network is a legitimate goal. For example, the following common analyses examine only the network structure: finding communities, fitting power-law graph models, enumerating motifs, measuring diffusion, and assessing resiliency [20]. While most of our discussion focuses on unlabeled graphs, we do study the impact of combining attributes and structural information in Section 5.

Whether a re-identification attack succeeds depends on two factors: the descriptive power of the adversary's external information and the structural similarity of nodes. Descriptive external information allows the adversary to distinguish between entities, allowing more accurate re-identification. But structural similarity means that entities can be hidden in a crowd, resisting re-identification.

To investigate these two factors, and to improve the anonymity of published networks, we make the following contributions:

- We propose three models of external information used by an adversary to attack naively-anonymized networks. These models represent a range of structural information that may be available to an adversary including complete and partial descriptions of node neighborhoods, and connections to hubs in the network. We formalize the structural indistinguishability of a node with respect to an adversary with locally-bounded external information. (Sections 2 and 3)

- We evaluate the effectiveness of these attacks in two ways. First, we apply the attacks to real networks from different domains, measuring successful node disclosures. Second, we study anonymity in random graphs. Our results show that real networks are diverse in their resistance to attacks, that anonymity is determined in part by a graph's density and degree distribution, and that hubs, while distinctive themselves, cannot be used to re-identify many of their neighbors. (Sections 4 and 5)

- To resist re-identification attacks, we propose a novel algorithm which anonymizes the graph by partitioning the nodes and then describing the graph at the level of partitions. The output is a generalized graph, which consists of a set of *supernodes* — one for each partition — and a set of *superedges* — which report the density of edges (in the original graph) between the partitions they connect. The generalized graph can be used to study graph properties by randomly sampling a graph that is consistent with the generalized graph description and then performing a standard analysis on this synthetic graph. These sampled graphs retain key properties of the original — such as degree distribution, path lengths, and transitivity — allowing complex analyses, such as network resiliency and disease transmission, to be accurately performed. (Section 6)

2. ANONYMITY IN NETWORKS

Before formally describing adversary knowledge in Section 3 we consider the practical properties of adversary knowledge that motivate our definitions. We also explain how structural similarity in a graph can protect against re-identification.

2.1 Knowledge Acquisition in Practice

Accurately modeling adversary knowledge is crucial for understanding the vulnerabilities of naively-anonymized networks, and for developing new anonymization strategies. External information about a published social network may be acquired through malicious actions by the adversary or from public information sources. In addition, a participant in the network, with some innate knowledge of entities and their relationships, may be acting as an adversary in an attempt to uncover unknown information. A legitimate privacy objective in some settings is to publish a network in which participating individuals cannot re-identify themselves.

Our goal is to develop parameterized and conservative models of external information that capture the power of a range of adversaries, and to then study the threats to anonymity that result. One of our guiding principles is that adversary knowledge about a targeted individual tends to be local to the targeted node, with more powerful adversaries capable of exploring the neighborhood around a node with increasing diameter. For the participant-adversary, whose knowledge is based on their participation in the network, existing research about institutional communication networks suggests that there is a horizon of awareness of about distance two around most individuals [11]. We formalize the external information available to an adversary through a set of knowledge queries described in the next section. Each knowledge query is parameterized by the radius around the targeted individual which it describes.

We also consider the impact of *hubs*, which are highly connected nodes observed in many networked data sets. In a Web graph, a hub may be a highly visited website. In a graph of email connections, hubs often represent influential individuals. Because hubs are often outliers in a graph's degree distribution, the true identity of hub nodes is often apparent in a naively-anonymized graph. In addition, an individual's connections to hubs may be publicly known or easily deduced. We consider attackers who use hub connections as a structural fingerprint to re-identify nodes.

Our assumption throughout the present work is that external information sources are accurate, but not necessarily complete. Accuracy means that when an adversary learns facts about a named individual, those facts are true of the original graph. However, we distinguish between a *closed-world* adversary, in which absent facts are false, and an *open-world* adversary in which absent facts are simply unknown. For example, when a closed-world adversary learns that Bob has three neighbors, he also learns that Bob has no more than three neighbors. An open-world adversary would learn only that Bob has at least three neighbors. Hub fingerprints have an analogous open- and closed-world interpretation.

In practice, an adversary may acquire knowledge that is complete. For example, an attacker who acquires the address book for a targeted individual would learn a complete list of their neighbors in an email communication network. Another example of closed-world external information is the attack proposed by Backstrom et al. [4]. They propose an attack in which a small group of participants in the network collude and each member of the group reveals all their relationships with all other participants.

As we would expect, closed-world adversaries are significantly more powerful (see the experimental results on real networks in Section 4). However, in many settings, the adversary cannot be certain that their information is complete and must assume an open world. We believe both closed- and open-world variants of adversary knowledge are important.

2.2 Anonymity Through Structural Similarity

Intuitively, nodes that look structurally similar may be indistinguishable to an adversary, in spite of external information. A strong form of structural similarity between nodes is *automorphic equivalence*. Two nodes $x, y \in V$ are automorphically equivalent (denoted $x \equiv_A y$) if there exists an isomorphism from the graph onto itself that maps x to y.

EXAMPLE 2.1. *Fred and Harry are automorphically equivalent nodes in the graph of Figure 1. Bob and Ed are not automorphically equivalent: the subgraph around Bob is different from the subgraph around Ed and no isomorphism proving automorphic equivalence is possible.*

Automorphic equivalence induces a partitioning on V into sets whose members have identical structural properties. It follows that an adversary — even with exhaustive knowledge of a target node's structural position — cannot identify an individual beyond the set of entities to which it is automorphically equivalent. We say that these nodes are *structurally indistinguishable* and observe that nodes in the graph achieve anonymity by being "hidden in the crowd" of its automorphic class members.

Some special graphs have large automorphic equivalence classes. For example, in a complete graph, or in a graph which forms a ring, all nodes are automorphically equivalent. But in most graphs we expect to find small automorphism classes, likely to be insufficient for protection against re-identification.

Though automorphism classes may be small in real networks, automorphic equivalence is an extremely strong notion of structural similarity. In order to distinguish two nodes in different automorphic equivalence classes, it may be necessary to use complete information about their positions in the graph. For example, for a weaker adversary, who only knows the degree of targeted nodes in the graph, Bob and Ed are indistinguishable (even though they are not automorphically equivalent). Thus we must consider the distinguishability of nodes to realistic adversaries with limited external information.

3. ADVERSARY KNOWLEDGE

We model the adversary's external information as access to a source that provides answers to a restricted *knowledge query* Q evaluated for a single target node of the original graph G. We always assume knowledge gathered by the adversary is accurate: that is, no spurious answers are provided to the adversary by the information source.

For a target node x, the adversary uses $Q(x)$ to refine the feasible candidate set. Since G_a is published, the adversary can easily evaluate *any* structural query directly on G_a. Thus the adversary will compute the refined candidate set that contains all nodes in the published graph G_a that are consistent with answers to the knowledge query on the target node.

DEFINITION 1 (CANDIDATE SET UNDER Q). *For a query Q over a graph, the* candidate set of x w.r.t Q *is* $cand_Q(x) = \{y \in V_a \mid Q(x) = Q(y)\}$.

In the next subsections we present three variants of adversary knowledge. The first is a class of very expressive structural queries that provide a precise way to capture structural knowledge of increasing diameter around a node, and that model an adversary with closed-world information about node degree. In Section 3.2, we present a less powerful class of queries, intended to model a weaker adversary who explores the graph edge-by-edge, possessing only open-world information. Lastly we consider knowledge provided by hub connections.

3.1 Vertex Refinement Queries

We define a class of queries, of increasing power, which report on the local structure of the graph around a node. These queries are inspired by iterative vertex refinement, a technique originally developed to efficiently test for the existence of graph isomorphisms [8]. The weakest knowledge query, $\mathcal{H}_0$, simply returns the label of the node. (We consider here unlabeled graphs, so $\mathcal{H}_0$ returns ϵ on all input nodes; these queries are extended to include attributes in Section 5.) The queries are successively more descriptive: $\mathcal{H}_1(x)$ returns the degree of x, $\mathcal{H}_2(x)$ returns the multiset of each neighbors' degree, and so on. The queries can be defined iteratively, where $\mathcal{H}_i(x)$ returns the multiset of values which are the result of evaluating $\mathcal{H}_{i-1}$ on the set of nodes adjacent to x:

$$\mathcal{H}_i(x) = \{\mathcal{H}_{i-1}(z_1), \mathcal{H}_{i-1}(z_2) \ldots, \mathcal{H}_{i-1}(z_m)\}$$

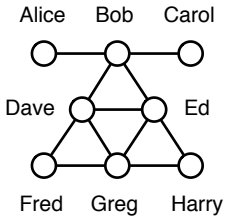
where $z_1 \ldots z_m$ are the nodes adjacent to x.

EXAMPLE 3.1. *Figure 2 contains the same graph from Figure 1 along with the computation of $\mathcal{H}_0$, $\mathcal{H}_1$, and $\mathcal{H}_2$ for each node. For example: $\mathcal{H}_0$ is uniformly ϵ. $\mathcal{H}_1(Bob) = \{\epsilon, \epsilon, \epsilon, \epsilon\}$, which we abbreviate in the table simply as 4. Using this abbreviation, $\mathcal{H}_2(Bob) = \{1, 1, 4, 4\}$ which represents Bob's neighbors' degrees.*

For each query $\mathcal{H}_i$, we define an equivalence relation on nodes in the graph in the natural way.

DEFINITION 2 (RELATIVE EQUIVALENCE). *Two nodes x, y in a graph are* equivalent relative to $\mathcal{H}_i$, *denoted $x \equiv_{\mathcal{H}_i} y$, if and only if $\mathcal{H}_i(x) = \mathcal{H}_i(y)$.*

EXAMPLE 3.2. *Figure 2(c) lists the equivalence classes of nodes according to relations $\equiv_{\mathcal{H}_0}, \equiv_{\mathcal{H}_1}$, and $\equiv_{\mathcal{H}_2}$. All nodes are equivalent relative to $\mathcal{H}_0$ (for an unlabeled graph). As i increases, the values for $\mathcal{H}_i$ contain successively more precise structural information about the node's position in the graph, and as a result, equivalence classes are divided.*



(a) graph

Node ID	$\mathcal{H}_0$	$\mathcal{H}_1$	$\mathcal{H}_2$
Alice	ϵ	1	$\{4\}$
Bob	ϵ	4	$\{1,1,4,4\}$
Carol	ϵ	1	$\{4\}$
Dave	ϵ	4	$\{2,4,4,4\}$
Ed	ϵ	4	$\{2,4,4,4\}$
Fred	ϵ	2	$\{4,4\}$
Greg	ϵ	4	$\{2,2,4,4\}$
Harry	ϵ	2	$\{4,4\}$

(b) vertex refinements

Equivalence Relation	Equivalence Classes
$\equiv_{\mathcal{H}_0}$	$\{A,B,C,D,E,F,G,H\}$
$\equiv_{\mathcal{H}_1}$	$\{A,C\}\quad\{B,D,E,G\}\quad\{F,H\}$
$\equiv_{\mathcal{H}_2}$	$\{A,C\}\{B\}\{D,E\}\{G\}\{F,H\}$
$\equiv_A$	$\{A,C\}\{B\}\{D,E\}\{G\}\{F,H\}$

(c) equivalence classes

Figure 2: (a) A sample graph, (b) external information consisting of vertex refinement queries $\mathcal{H}_0, \mathcal{H}_1$ and $\mathcal{H}_2$ computed for each individual in the graph, (c) the equivalence classes of nodes implied by vertex refinement. For the sample data, $\equiv_{\mathcal{H}_2}$, corresponds to automorphic equivalence, $\equiv_A$.

To an adversary limited to knowledge query $\mathcal{H}_i$, nodes equivalent with respect to $\mathcal{H}_i$ are indistinguishable. The following proposition formalizes this intuition:

PROPOSITION 1. *Let $x, x' \in V$. If $x \equiv_{\mathcal{H}_i} x'$ then $cand_{\mathcal{H}_i}(x) = cand_{\mathcal{H}_i}(x')$.*

Iterative computation of $\mathcal{H}$ continues until no new vertices are distinguished. We call this query $\mathcal{H}^*$. In the example of Figure 2, $\mathcal{H}^* = \mathcal{H}_2$. The vertex refinement technique is the basis of efficient graph isomorphism algorithms which can be shown to work for almost all graphs [3]. In our setting, this means that equivalence under $\mathcal{H}^*$ is very likely to coincide with automorphic equivalence. In Section 5 we analyze theoretically the expected disclosure under $\mathcal{H}_i$ for various random graph models and graph densities.

3.2 Subgraph Queries

Vertex refinement queries are a concise way to describe locally expanding structural queries. However, as a model of adversary knowledge they have two limitations. First, they always provide complete information about the nodes adjacent to the target (they are therefore an instance of closed-world knowledge). For example, $\mathcal{H}_1(x)$ returns the *exact* number of neighbors of x. Second, $\mathcal{H}$ queries can describe arbitrarily large subgraphs around a node if that node is highly connected. For example, if $\mathcal{H}_1(x) = 100$, the adversary learns about a large subgraph in G, whereas $\mathcal{H}_1(y) = 2$ provides much less information. Thus, the index of the $\mathcal{H}$ query may be a coarse measure of the amount of information learned.

As an alternative, we consider a very general class of queries which assert the existence of a subgraph around the target node. We measure the descriptive power of a query by counting the number of edges in the described subgraph; we refer to these as *edge facts*. For example, Figure 3 illustrates three subgraph queries centered around *Bob*. The first simply asserts that *Bob* has (at least) three distinct neighbors, the second describes a tree of nodes near *Bob*, and the third relates nearby nodes in a subgraph. These informal query patterns use 3, 4, and 5 edge facts, respectively.

Note that we do not model an adversary capable of constructing and evaluating arbitrary subgraph queries. Instead, we assume the adversary is capable of gathering some fixed number of edge facts around the target x. The adversary learns the existence of a subgraph around x which may be incomplete. The existence of this subgraph can be expressed as a query, and we model the adversary's knowledge by granting the answer to such a query. Because such a query has existential semantics, it is open-world knowledge.

Naturally, for a fixed number of edge facts there are many subgraph queries that are true around a node x. These can be thought of as corresponding to different strategies of knowledge acquisition that could be employed by the adversary. In testing the distinguishing power of subgraph queries in Section 4, we test a range of strategies including breadth-first exploration, induced subgraphs of radius 1 and 2, and strategies that emphasize small distinctive structures. For a given number of edge facts, some queries are more effective at distinguishing individuals. We report on the diversity of disclosure that can result from a fixed number of edge facts.

3.3 Hub Fingerprint Queries

A hub is a node in a network with high degree and high betweenness centrality (the proportion of shortest paths in the network that include the node). In an email communication graph, a hub may correspond to a well-known administrator responsible for disseminating information to all department members. Hubs are important components of the topology of networks, and they have been widely observed in social and information networks [20]. Hubs are often outliers in a network, making it difficult to protect their identity through anonymization. For example, in a naively-anonymized network trace, the hubs correspond to the most frequently visited websites, which are typically known by an adversary.

A hub fingerprint for a target node x is a description of the node's connections to a set of designated hubs in the network. We denote the hub fingerprint of x by $\mathcal{F}_i(x)$ where the subscript i places a limit on the maximum distance of observable hub connections. For example, if we consider Dave and Ed hubs, Fred's hub fingerprint is a vector of his shortest path lengths (bounded by i) to each hub. $\mathcal{F}_1(Fred) = (1,0)$ because Fred is distance 1 from Dave but not connected to Ed in one hop or less; $\mathcal{F}_2(Fred) = (1,2)$ because Fred is distance 1 from Dave and distance 2 from Ed.

We consider an adversary capable of gathering hub fingerprints in both an open and a closed world. In the closed world, the lack of a connection to a hub implies with certainty that no connection exists. In the open world, the absence of a connection in a hub fingerprint may simply represent incompleteness in the adversary's knowledge. Thus in the open world, if the adversary knows $\mathcal{F}_1(Fred) = (1,0)$ then nodes in the anonymized graph with $\mathcal{F}_1$ fingerprints of $(1,0)$ or $(1,1)$ are both candidates for Fred. In Section 4, we study the distinguishing power of hub fingerprints empirically by computing candidate sets in real networks.

3.4 Comparison of Knowledge Models

Vertex refinement queries and subgraph queries are related, but they differ in their expressiveness and the efficiency of evaluating their re-identification risk. First, vertex refinement queries provide complete information about node degree. A subgraph query can

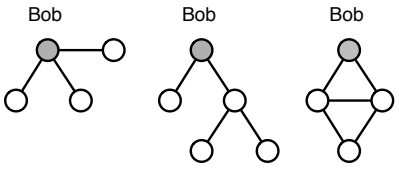


Figure 3: Three instances of the partial information about entity *Bob* that can be expressed as a subgraph query.

Table 1: Summary of networks studied.

Statistic	Data Set		
	Hep-Th	Enron	Net-trace
Nodes	2510	111	4213
Edges	4737	287	5507
Minimum degree	1	1	1
Maximum degree	36	20	1656
Median degree	2	5	1
Average degree	3.77	5.17	2.61
Avg. cand. set size ($\mathcal{H}_1$)	558.45	12.05	2792.09
Avg. cand. set size ($\mathcal{H}_2$)	25.38	1.49	608.58
Fraction re-identified ($\mathcal{H}_1$)	0.002	0.027	0.006
Fraction re-identified ($\mathcal{H}_2$)	0.404	0.739	0.111

never express $\mathcal{H}_i$ knowledge because subgraph queries are existential and cannot assert exact degree constraints or the absence of edges in a graph. Second, the complexity of computing $\mathcal{H}^*$ is linear in the number of edges in the graph, and is therefore efficient even for large datasets. Evaluating subgraph queries, on the other hand, can be NP-hard in the number of edge facts, as computing candidate sets for subgraph queries requires finding all isomorphic subgraphs in the input graph. Although we do not place computational restrictions on the adversary, the vertex refinement queries allow a data owner to efficiently assess disclosure risk.

Yet, the semantics of subgraph queries seem to model realistic adversary capabilities more accurately. It may be difficult for an adversary to acquire the detailed structural description of higher-order vertex refinement queries. Nevertheless, we believe vertex refinement queries offer an efficient and conservative measure of structural diversity in a graph. In addition, $\mathcal{H}_i$ queries are conceptually appealing as they represent a natural spectrum of structural knowledge, beginning with $\mathcal{H}_1$ which reports node degree, and converging, as i increases, on automorphic equivalence.

Finally, we note that both of the above models of knowledge have well-studied logical foundations. $\mathcal{H}_i$ knowledge corresponds to first order logic with counting quantifiers, restricted to i variables [15]. Subgraph queries can be expressed as conjunctive queries with disequalities. The number of edge facts corresponds to the number of subgoals in the query.

4. DISCLOSURE IN REAL NETWORKS

In this section we evaluate empirically the impact of external information on the adversary's ability to re-identify individuals.

We study three networked data sets, drawn from diverse domains. For each data set, we consider each node in turn as a target. We assume the adversary computes a vertex refinement query, a subgraph query, or a hub fingerprint query on that node, and then we compute the corresponding candidate set for that node. We report the distribution of candidate set sizes across the population of nodes to characterize how many nodes are protected and how many are identifiable.

We use the following data sets. The **Hep-Th** database describes papers and authors in theoretical high-energy physics, taken from

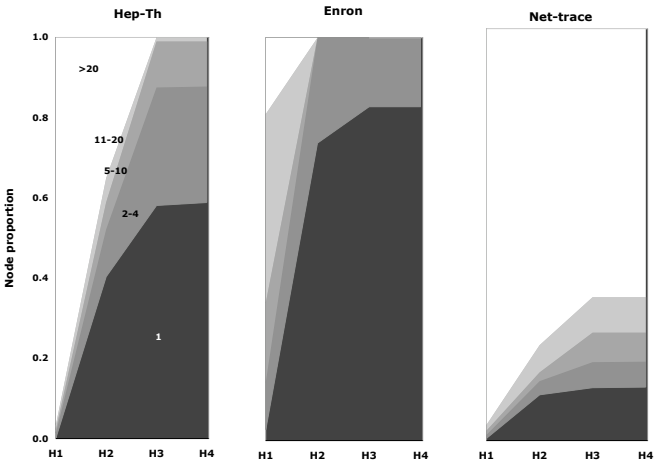


Figure 4: Relationship between candidate size and vertex refinement knowledge $\mathcal{H}_i$ for $i = 1..4$ for three network datasets. The trend lines show the percentage of nodes whose candidate sets have sizes in the following buckets: $[1]$ (black), $[2, 4]$, $[5, 10]$, $[11, 20]$, $[21, \infty]$ (white).

the arXiv archive. We extracted a subset of the authors and considered them linked if they wrote at least two papers together. The **Enron** dataset is derived from a corpus of email sent to and from managers at Enron Corporation, made public by the Federal Energy Regulatory Commission during its investigation of the company. Two individuals are connected if they corresponded at least 5 times. The **Net-trace** dataset was derived from an IP-level network trace collected at a major university. The trace monitors traffic at the gateway; it produces a bipartite graph between IP addresses internal to the institution, and external IP addresses. We restricted the trace to 187 internal addresses from a single campus department and the 4026 external addresses to which at least 20 packets were sent on port 80 (http traffic).

All datasets have undirected edges, with self-loops removed. We eliminated a small percentage of disconnected nodes in each dataset, focusing on the largest connected component in the graph. Detailed statistics for the datasets are shown in Table 1.

4.1 Re-identification: Vertex Refinement

Recall from Section 3 that nodes contained in the same candidate set for knowledge $\mathcal{H}_i$ share the same value for $\mathcal{H}_i$, are indistinguishable according to $\mathcal{H}_i$, and are therefore protected if the candidate set size is sufficiently large.

Figure 4 is an overview of the likelihood of re-identification under $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ and $\mathcal{H}_4$ knowledge queries. For each $\mathcal{H}_i$, the graph reports on the percentage of nodes whose candidate sets have sizes in the following buckets: $[1]$, $[2, 4]$, $[5, 10]$, $[11, 20]$, $[21, \infty]$. Nodes with candidate set size 1 have been uniquely identified, and nodes with candidate sets between 2 and 4 are at high risk for re-identification. Nodes are at fairly low risk for re-identification if there are more than 20 nodes in their candidate set.[1] Each $\mathcal{H}_i$ is represented as a different point on the x-axis.

[1]We do not suggest these categories as a universal privacy standard, but merely as divisions that focus attention on the most important part of the candidate set distribution where serious disclosures are at risk.

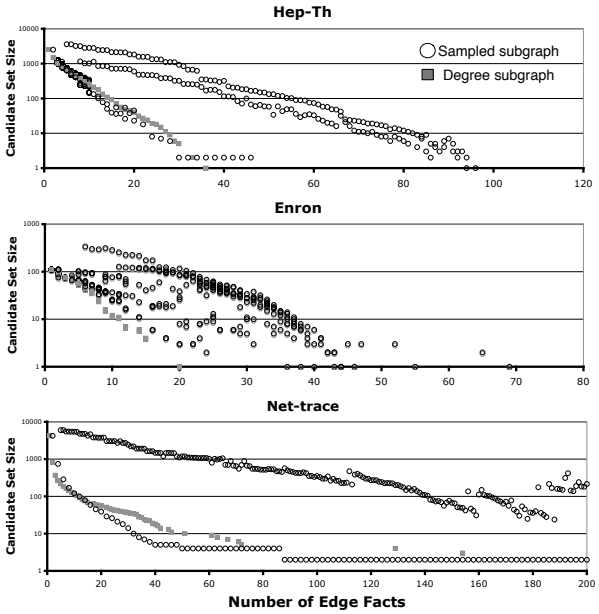


Figure 5: Candidate set sizes (on a log scale) for sampled subgraph queries consisting of specified number of edge facts. (Please note differences in scale.)

Figure 4 shows that for the **Hep-Th** data, $\mathcal{H}_1$ leaves nearly all nodes at low risk for re-identification, and it requires $\mathcal{H}_3$ knowledge to uniquely re-identify a majority of nodes. For **Enron**, under $\mathcal{H}_1$ about 15% of the nodes have candidate sets smaller than 5, while only 19% are protected in candidate sets greater than 20. Under $\mathcal{H}_2$, re-identification jumps dramatically so that virtually all nodes have candidate sets less than 5.

Net-trace has substantially lower disclosure overall, with very few identified nodes under $\mathcal{H}_1$, and even $\mathcal{H}_4$ knowledge does not uniquely identify more than 10% of the nodes. This results from the unique bipartite structure of the network trace dataset: many nodes in the trace have low degree, as they are unique or rare web destinations contacted by only one internal host.

A natural precondition for publication is a very low percentage of high-risk nodes under a reasonable assumption about adversary knowledge. Two datasets meet that requirement for $\mathcal{H}_1$ (**Hep-Th** and **Net-trace**), but no datasets meet that requirement for $\mathcal{H}_2$.

Overall, we observe that there can be significant variance across different datasets in their vulnerability to different adversary knowledge. However, across all datasets, the most significant change in re-identification is from $\mathcal{H}_1$ to $\mathcal{H}_2$, illustrating the increased power of adversaries that can explore beyond the target's immediate neighborhood. Re-identification tends to stabilize after $\mathcal{H}_3$—more information in the form of $\mathcal{H}_4$ does not lead to an observable increase in re-identification in any dataset. Finally, even though there are many re-identified nodes, a substantial number of nodes are *not* uniquely identified even with $\mathcal{H}_4$ knowledge.

4.2 Re-identification: Subgraph Queries

Recall from Section 3 that we also model an adversary exploring the local graph around a target individual, and we measure that knowledge by counting the number of edge facts acquired. Figure 5 shows the relationship between the number of edge facts and re-identification success. Each point represents a subgraph query of a specified size; the re-identification success is measured by the size of the candidate set (vertical axis). For a fixed number of edge facts, there are many possible subgraph queries. We simulated adversaries who gather facts around the target according to a variety of strategies: breadth-first exploration (labelled "Degree subgraphs" in the figure), random subgraphs, induced subgraphs of radius 1 and 2, and small dense structures (collectively referred to as "Sampled subgraphs").

Overall, disclosure is substantially lower than for vertex refinement queries. To select candidate sets of size less than 10 requires a subgraph query of size 24 for **Hep-Th**, size 12 for **Enron**, and size 32 for **Net-trace**. The smallest subgraph query resulting in a unique disclosure was size 36 for **Hep-Th** and size 20 for **Enron**. The smallest candidate set witnessed for **Net-trace** was size 2, which resulted from a query consisting of 88 edge facts.

Breadth-first exploration led to selective queries across all three datasets. Such a query explores all neighbors of a node and then starts to explore all neighbors of a randomly chosen neighbor, etc. This asserts lower bounds on the degree of nodes. In **Enron**, these were the most selective subgraph queries witnessed; for **Hep-Th** and **Net-trace**, the more selective subgraph queries asserted the existence of two nodes with a large set of common neighbors.

The results presented above illustrate the diverse subset of subgraph queries we sampled. While it is clearly intractable to perform an exhaustive search over all possible subgraphs and matching them to each node in the graph, it is an interesting open question to determine, given a graph, and a fixed number of edge facts, the subgraph query that will result in the smallest candidate set. This would reflect the worst-case disclosure possible from an adversary restricted to a specified number of edge facts. Finding the worst-case subgraph queries is related to searching for motifs in a network [20], which are small structures that occur frequently. Efficient algorithms for this problem are currently under investigation [13].

4.3 Re-identification: Hub Fingerprints

Recall that the hub fingerprint query $\mathcal{F}_i(x)$ returns a vector describing the length of the shortest path from x to each of a distinguished set of hubs. In Figure 6 we show the candidate set sizes for hub fingerprints $\mathcal{F}_1$ and $\mathcal{F}_2$, choosing the five highest degree nodes as hubs for **Enron**, and the ten highest degree nodes for both **Hepth** and **Net-trace**. The choice of the number of hubs was made by considering whether the degree of the node was distinguishable in the degree distribution and therefore likely to be an outlier in the original graph. We computed the candidate sets for these hub fingerprint queries under both the closed-world interpretation and the open-world interpretation.

Generally we find disclosure is low using hub fingerprints. At distance 1, 54% of the nodes in **Enron** were not connected to any hub and therefore hub fingerprints provide no information. This statistic was 90% for **Hepth** and 28% for **Net-trace**. In addition, connectivity to hubs was fairly uniform across individuals. For example, the space of possible fingerprints at distance 1 for **Hepth** and **Net-trace** is $2^{10} = 1024$. Of these, only 23 distinct fingerprints were observed for **Hepth** and only 46 for **Net-trace**.

In essence, there are two competing effects on the distinguishing power of hubs. While hubs themselves stand out, they have high-degrees, which means connections to a hub are shared by many. While hubs would appear to be a challenge for anonymization, this finding suggests that disguising hubs in published data may not be required to maintain anonymity.

5. ANONYMITY IN RANDOM GRAPHS

In this section, we study re-identification risk by analyzing mod-

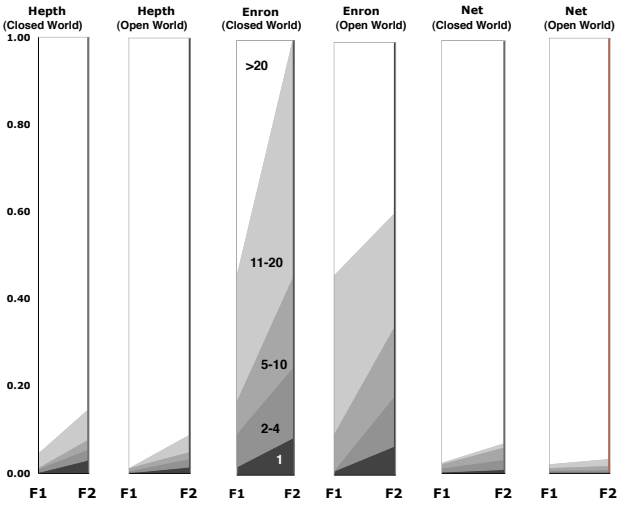


Figure 6: Candidate set sizes for hub fingerprint queries: $\mathcal{F}_1$ and $\mathcal{F}_2$ are shown for each dataset under a closed-world and open-world assumption.

els of random graphs. While a thorough theoretical study of graph models and anonymity risk is a subject of future work, we provide here initial results that describe some of the most important relationships between graph models, their key properties, and our models of knowledge.

First we use $\mathcal{H}_i$ knowledge to study re-identification risk in two popular models of graphs: classical random graphs and power-law graphs. Second, we describe a result which allows us to guarantee protection against subgraph queries of a specific size based on the presence of cliques. Last, we consider random graphs with attributes and show re-identification success is determined by attribute-structure correlation.

5.1 Re-identification in Random Graphs

Here, we study how anonymity is affected by two key graph properties, density and degree distribution. To study the relationship between graph density and anonymity, we analyze the Erdős-Rényi (ER) model, the simplest random graph model. Following that, we study random graphs with power-law degree distributions.

The ER model generates a graph with n nodes by sampling each edge independently with probability p. As the number of nodes, n, increases, these graphs exhibit different behaviors depending on how p scales as a function of n. Three scalings correspond to *sparse*, $p = c/n$, *dense*, $p = c \log n/n$, and *super-dense*, $p = c$ (where c is a constant). The first two are of interest because when $c > 1$, the graph includes a giant connected component of size $\Theta(n)$ and a collection of smaller components (in the sparse case) or the graph is completely connected (in the dense case) [10]. We consider below the sparse and super-dense cases and conclude with some remarks regarding the dense case.

We begin with sparse graphs. In sparse graphs, nodes cannot be distinguished because the graph lacks sufficient edge density to create diversity in structure. Because the edge probability is $p = c/n$, the expected node degree, which is $p(n-1)$, goes to c as $n \to \infty$. Intuitively, because the expected degree is constant, then for sufficiently large n, structural patterns must repeat, leading to structural uniformity. As the following theorem shows, no degree of $\mathcal{H}_i$ knowledge will be distinguishing. (See the Appendix for theorem proofs.)

THEOREM 1 (SPARSE ER RANDOM GRAPHS). *Let G be an ER random graph containing n nodes with edge probability given by $p = c/n$ for $c > 1$. With probability going to one, the expected sizes of the equivalence classes induced by $\mathcal{H}_i$ is $\Theta(n)$, for any $i \geq 0$.*

This is an encouraging result for large graphs. (However, in simulations, we found that some re-identification occurs in random graphs of less than 10^6 nodes.) We now consider the case of a super-dense graph where $p = 1/2$. The following theorem, originally due to Babai and Kucera [3] but rephrased here, shows that with high probability, every node will be uniquely identified using $\mathcal{H}_3$ knowledge:

THEOREM 2 (SUPER-DENSE ER RANDOM GRAPHS). *Let G be an ER random graph on n nodes with edge probability $p = 1/2$. The probability that there exist two nodes $x, y \in V$ such that $x \equiv_{\mathcal{H}_3} y$ is less than 2^{-cn} for constant value $c > 0$.*

This result provides a sufficient condition for unique re-identification of the entire population in a network. It is more disappointing from a privacy perspective than the previous result. Fortunately, few real graphs exhibit such a high average degree and ER graphs with edge probability $p = 1/2$ are not realistic models of the networks analysts are likely to study as most social and communication networks tend to be sparse. Thus Theorem 1 is likely to be more applicable.

In between these densities, there is the class of dense ER graphs where $p = c \log n/n$ and the expected degree is $c \log n$. The case where $c > 1$ is of particular interest because it gives rise to connected graphs for sufficiently large n. These dense graphs better match realistic expectations of graph growth since the average degree grows slowly as the size of the population increases. Preliminary analysis, coupled with simulation, suggests that, for sufficiently large n, nodes cannot be identified by $\mathcal{H}_1$ for any $c > 0$ but, unfortunately, that all nodes are re-identified by $\mathcal{H}_2$ for any $c > 1$.

Unlike ER graphs, real networks often have degree distributions which follow a power law (are heavy-tailed). To capture this property, several graph models have been proposed, including the power law random graph (PLRG) model [1]. In this model, a graph is constructed by first assigning a degree to each node, where the degree is sampled from a power law distribution. Edges are inserted by randomly choosing endpoints until every node has as many edges as its specified degree. (This can result in self-loops or multiple edges between a pair of nodes.)

The PLRG, and other power-law models, generate graphs with constant average degree as the number of nodes increases. Thus the edge density is low, and despite the skew in node degree, the structural diversity is insufficient for re-identification. We state this formally for PLRG because it is the easiest model to analyze.

THEOREM 3 (POWER-LAW RANDOM GRAPHS). *Let G be a PLRG on n nodes. With probability going to one, the expected sizes of the equivalence classes induced by $\mathcal{H}_i$ is $\Theta(n)$, for any $i \geq 0$.*

Simulations show that the same relationship holds for other power-law models, such as the preferential attachment model [5].

5.2 Anonymity Against Subgraph Queries

The existence of a clique in a graph immediately implies re-identification resistance against any subgraph query that can be embedded in the clique. For a graph G, its clique number, denoted $\omega(G)$, is the number of nodes in the largest clique.

PROPOSITION 2. *Let G be any graph, and $Q(x)$ a subgraph query around any node x. If $Q(x)$ contains fewer than $\omega(G)$ nodes, then $|cand_Q(x)| \geq \omega(G)$.*

This proposition holds because any subgraph query mentioning fewer than $\omega(G)$ nodes, irrespective of the number of edges, will match any node in the fully connected clique (recall that our subgraph queries assume an open-world). It will therefore have a candidate set of size at least $\omega(G)$. This proposition allows a data owner to quickly determine the minimum size subgraph that could possibly re-identify a node in their graph. For example, the clique number of both the **Hepth** and **Enron** data sets is seven.[2] Any subgraph query mentioning 7 or fewer nodes (which could contain as many as $7 * 6/2 = 21$ edges) will not succeed in distinguishing an individual to within 7 candidates.

Proposition 2 is also useful because the expected clique number for various graph models is known. For ER random graphs, the expected clique number is known to be approximately $2\log n$ for almost every graph [28]. Compared with ER random graphs, the cliques observed in many social and communication networks are substantially larger. Recently, Bianconi et al. calculated bounds on the expected clique number for models of scale-free graphs that match observed properties of some social networks [6]. These bounds, combined with Proposition 2, result in useful lower bounds on the disclosure under subgraph queries in open worlds.

5.3 Random Graphs with Attributes

Until now we have treated our social networks as unlabeled, focusing exclusively on structural re-identification. Clearly nodes in a social network may contain descriptive attributes relevant to social analysis. For example, age, gender, and salary might be common attributes in an institutional social network. If an adversary can discover the attributes of individuals known to be present in the data set, then these attributes can act as quasi-identifiers, in combination with structural features. To model knowledge of attributes and structure, we use vertex refinement queries where the initial query $\mathcal{H}_0$ returns a node label instead of ϵ. We use the notation $\mathcal{H}_i^A$ to denote vertex refinement queries augmented with attributes.

The degree to which attribute knowledge helps to re-identify individuals depends first, on the selectivity of published attributes, and second, on the correlation between attributes and structure. To explore these factors, we augment a sparse ER random graph ($|V| = 10,000$, $|E| = 20,000$) with randomly assigned attributes. Two parameters govern the assignment process: the number of distinct attribute values in the label alphabet A, $|A| \in [1, 10000]$; and the degree to which attribute assignment is correlated with structure, governed by parameter $p \in [0, 1]$. We explain the use of this correlation parameter later. First, we look at the affect of attributes generated independently from structure.

We find that attributes, when combined with structural information provided by vertex refinement queries, have a compounding effect on re-identification. Recall that $\mathcal{H}_0$ simply returns the label of a node without any information about its neighborhood. For an unlabeled graph, there is a single candidate set of size $|V|$ under $\mathcal{H}_0$. As expected, if attributes from an alphabet of size $|A|$ are added uniformly to nodes, then the average candidate set size for $\mathcal{H}_0^A$ diminishes to $|V|/|A|$. As i increases, this effect continues, each time dividing the average candidate size by at least a factor of $|A|$ and increasing re-identification. These results are shown in Figure 7 (top), and demonstrate the significant power of an adversary capable of discovering descriptive attributes of nodes along with structural properties.

[2]Since **Net-trace** is a bipartite graph, its clique number is 2.

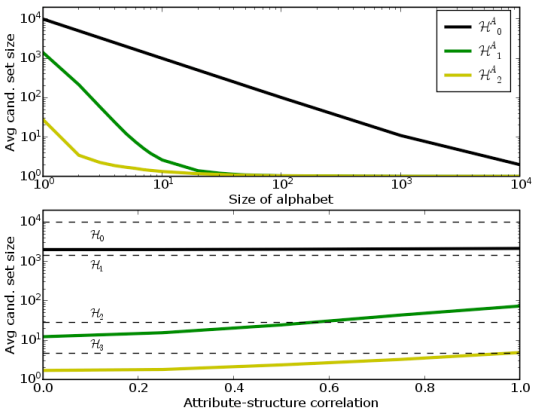


Figure 7: Effect of attribute knowledge on re-identification measured by average size of candidate sets. Top: attributes are sampled independently ($p = 0$) from an alphabet of varying size. Bottom: attribute-structure correlation p varies while alphabet size is fixed at five. The dashed lines show candidate set sizes under $\mathcal{H}_i$ (no attribute knowledge).

However, this analysis assumes that attributes are assigned to nodes independently of structure. Instead, the structural properties of nodes in a social network are often correlated with attributes of the nodes. This phenomenon dampens the increased re-identifiability possible with attributes. To sample attributes correlated with structure, we map each node degree to an attribute value by partitioning the degree distribution into $|A|$ bins. Then with probability p we assign a node to the attribute associated with its degree; with probability $(1 - p)$ we sample an attribute value uniformly at random. Observe that when $p = 0$, attributes are uniformly distributed. When $p = 1$, then attribute knowledge serves as approximate degree knowledge; as $|A|$ approaches the number of distinct degrees in the graph, the approximation becomes exact. Figure 7 (bottom) shows the increase in average candidate set size with increasing correlation. For example, when $p = 0$, $\mathcal{H}_1^A$ knowledge is more informative than $\mathcal{H}_2$; but as correlation increases it becomes less informative, being equally informative at around $p = 0.6$.

6. ANONYMIZATION ALGORITHM

In this section we describe an anonymization technique that protects against re-identification by generalizing the input graph. We generalize a graph by grouping nodes into partitions, and then publishing the number of nodes in each partition, along with the density of edges that exist within and across partitions. The adversary attempts re-identification in the generalized graph, while the analyst uses it to study properties of the original graph.

6.1 Graph Generalization

To generalize a naively-anonymized graph $G_a = (V_a, E_a)$, we partition its nodes into disjoint sets. The elements of a partitioning $\mathcal{V}$ are subsets of V_a. They can be thought of as *supernodes* since they contain nodes from G_a, but are themselves the nodes of a undirected generalized graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The *superedges* of $\mathcal{E}$ include self-loops and are labeled with non-negative weights by the function $d : \mathcal{E} \rightarrow Z^*$. $\mathcal{G}_\mathcal{V}$ is a generalization of G_a under a partitioning $\mathcal{V}$ if the edge labels report the density of edges (in G_a) that exist within and across the partitions:

DEFINITION 3 (GENERALIZATION OF GRAPH). *Let $\mathcal{V}$ be the supernodes of V_a. $\mathcal{G}$ is a* generalization of G_a under $\mathcal{V}$ *if, for all* $X, Y \in \mathcal{V}$, $d(X,Y) = |\{(x,y) \in E_a |\ x \in X, y \in Y\}|$.

$\mathcal{G}$ summarizes the structure of G_a, but the accuracy of that summary depends on the partitioning. For any generalization $\mathcal{G}$ of G_a, we denote by $\mathcal{W}(\mathcal{G})$, the set of possible worlds (graphs over V_a) that are consistent with $\mathcal{G}$. Intuitively, this set of graphs is generated by considering each supernode X and choosing exactly $d(X,X)$ edges between its elements, then considering each pair of supernodes (X,Y) and choosing exactly $d(X,Y)$ edges between elements of X and elements of Y. The size of $\mathcal{W}(\mathcal{G})$ is a measure of the accuracy of $\mathcal{G}$ as a summary of G_a.

The partitioning of nodes is chosen so that the generalized graph satisfies privacy goals and maximizes utility, as explained in Sections 6.2 and 6.3 respectively. In the extreme case that all partitions contain a single node, then the graph generalization $\mathcal{G}$ does not provide any additional anonymity: $\mathcal{W}(\mathcal{G})$ contains just the graph G_a (the function d encodes its adjacency matrix). At the other extreme, if all nodes are grouped into a single partition, then $\mathcal{G}$ consists of a single supernode with a self-loop labeled with $|E_a|$ (the total number of edges in the original graph). $\mathcal{W}(\mathcal{G})$ is thus the set of all graphs over V_a with $|E_a|$ edges. In this case the generalization provides anonymity, but is unlikely to be useful to the analyst since it reflects only the edge density of the original graph.

In studying a generalized graph, the analyst can sample a single random graph from $\mathcal{W}(\mathcal{G})$ and then perform standard graph analysis on this synthetic graph. Repeated sampling can improve the accuracy of analysis. We study in Section 6.4 the bias and variance of estimates of graph properties based on graphs sampled from $\mathcal{W}(\mathcal{G})$.

6.2 Anonymity of Generalized Graphs

To ensure anonymity we require that the adversary have a minimum level of uncertainty about the re-identification of any target node in V. Because only the edge density is published with each partition, it is never possible for the adversary to distinguish between individuals in a partition. Therefore, in reasoning about the candidate set for a query Q, the adversary can attempt to refine the candidates for a target node x only down to a set of feasible *partitions*, with each node in the partition an equally-likely candidate.

As a result, we use the size of a partition to provide a basic guarantee against re-identification, similar to that provided by k-anonymity, and require that each partition have size at least k. This ensures that $\text{cand}_Q(x) \geq k$ for any adversary Q. Unlike most formulations of k-anonymity, k is only a lower bound: some adversaries may not be able to re-identify the target node's partition. For example, consider an adversary who knows only the degree of its target. The adversary must consider as candidates the nodes of all partitions X for which $\text{mindegree}(X) \leq m \leq \text{maxdegree}(X)$ where mindegree and maxdegree are defined as follows. For any partition $X \in \mathcal{V}$ and any node $x \in X$, there exists some graph in $\mathcal{W}(\mathcal{G})$ where the degree of x is $\text{mindegree}(X) = \max(0, d(X,X) - \binom{|X|-1}{2}) + \sum_{Y \in \mathcal{V}} \max(0, d(X,Y) - (|X|-1)|Y|)$. Similarly, there is some graph in $\mathcal{W}(\mathcal{G})$ where the degree of x is as large as $\text{maxdegree}(X) = \min(|X|-1, d(X,X)) + \sum_{Y \in \mathcal{V}} \min(|Y|, d(X,Y))$ Thus, for some adversaries the candidate set may include multiple partitions and the lower bound on candidate set size may be even larger than k. However, capitalizing on this improved bound introduces additional complexity into the graph generalization algorithm because it must then obey an enhanced privacy constraint. We conservatively require k-sized partitions and leave improvements in this area as future work.

6.3 Algorithm Description

We now present the graph generalization algorithm. The input to the algorithm is G_a and privacy parameter k. The output is a partitioning $\mathcal{V}$ of V_a which determines the generalized graph $\mathcal{G}$ that is published.

Subject to the privacy constraint, which requires partitions of size at least k, we would like to find the partitioning that best fits the input graph. We estimate fitness via a maximum likelihood approach. We consider a uniform probability distribution over the possible worlds $\mathcal{W}(\mathcal{G})$. For a graph $g \in \mathcal{W}(\mathcal{G})$ we define $\Pr_{\mathcal{G}}[g] = 1/|\mathcal{W}(\mathcal{G})|$ where the number of possible worlds is:

$$|\mathcal{W}(\mathcal{G})| = \prod_{X \in \mathcal{V}} \binom{\frac{1}{2}|X|(|X|-1)}{d(X,X)} \prod_{X,Y \in \mathcal{V}} \binom{|X||Y|}{d(X,Y)}$$

Without regard to the anonymity condition, the partitioning that maximizes likelihood is the one with each node in a separate partition. Then, as explained above, $|\mathcal{W}(\mathcal{G})| = 1$ and $\Pr_{\mathcal{G}}[G_a] = 1$. In general, likelihood is greater with more partitions because each partition introduces more parameters to fit a fixed amount of data. But subject to the minimum size constraint, partitionings can vary greatly in their fit to the input graph. The algorithm uses local search to explore the exponential number of partitionings.

The design of the search algorithm is based on techniques for solving a related social network analysis problem: *stochastic block-modeling* [20]. The objective of stochastic block-modeling is to cluster the nodes of the graph so that nodes in the same group play a similar "social role" in the network. While the high-level idea is the same, there are a few key distinctions from our work. First, our differing motivations result in different likelihood functions. In stochastic block-modeling, the goal is to build a predictive model of the data and so the likelihood includes a penalty term for model complexity; in contrast, our goal is to fit the original network as closely as possible given the anonymity condition. Second, the anonymity condition imposes a new constraint on the search space, which makes search more complex.

To find the partitioning that maximizes the likelihood function, the algorithm searches using simulated annealing [23]. Each valid partitioning (i.e., a minimum partition of at least k nodes) is a state in the search space. Starting with a single partition containing all nodes, the algorithm proposes a change of state, by splitting a partition, merging two partitions, or moving a node to a different partition. The proposal of moving from partition $\mathcal{V}$ to some new partition $\mathcal{V}'$ is evaluated based on the change in likelihood that results. The proposal is always accepted if it improves the likelihood and accepted with some probability if it decreases the likelihood. The acceptance probability starts high and is cooled slowly until, as it approaches zero, a move is accepted only if it increases the likelihood. We terminate search when fewer than 10% of proposals are accepted.

The algorithm may return a partitioning that is only locally maximal. Whether this happens depends in part on the cooling schedule of simulated annealing; if cooled slowly enough, it will return the global maximum with high probability [23]. Nevertheless, finding the globally optimal partition is an intractable problem, and we cannot quantify how close the output is to the optimum. In experimental results not shown, we did a more systematic exploration of the search space using random restarts. On the **Enron** graph with $k = 3$, the log-likelihood of the output partition ranged from -362.6 to -353.3; in contrast, a greedy algorithm returns a partition with log-likelihood of only -511.5.

To make search more efficient, we cache the statistics needed to compute likelihood. We maintain a cache of edge counts $d(X,Y)$

to facilitate computing the likelihood. Furthermore, when considering a move in search space, it is only necessary to compute the change in likelihood, which is more efficient since a move only affects a subset of terms in the likelihood equation. For example, to split assignment X into X' and X'', the only affected terms are the ones involving X. Furthermore, it is only necessary to consider those Y where $d(X, Y) > 0$. Since the input graphs are sparse, there are typically only a small number of affected terms. In the worst-case, computing the change in likelihood requires time that is linear in the size of the input graph.

We also made a few design choices that make search more efficient. Partitions are split in a greedy fashion: a randomly chosen node is moved from X to a new group X', and then for each of the next $k - 1$ nodes, we select the node that maximizes the likelihood when moved from X to X'. Second, for merges and node moves, we only consider partitions X, Y where $d(X, Y) > 0$ or there exists Z such that $d(X, Z) > 0$ and $d(Y, Z) > 0$. This is locally optimal, in that if Y does not satisfy this condition, then merging X and Y can only decrease the likelihood of the current partitioning. While these choices may exclude the optimal assignment, results indicate that they are effective heuristics: they greatly reduce runtime without any decrease in likelihood.

Based on experiments on the three networks, search terminates after roughly $100\,|V|$ steps. Since each step takes worst-case linear time, the total runtime of the algorithm is $O(n^2)$. For the smaller dataset, **Enron**, search completes in minutes; for the largest network, **Net-trace**, search plateaus after a few hours.

6.4 Experimental Results

We now investigate how graph generalization affects network properties. The algorithm output is a generalized graph $\mathcal{G}$, which the analyst can use to infer properties of the original graph. Here, we consider an analyst who estimates a graph property by drawing samples graphs from $\mathcal{W}(\mathcal{G})$, measuring the property of each sample, and then aggregating measurements across samples. Anonymization can introduce two sources of error: estimates of a graph property can be systematically biased or highly variable. We investigate the bias and variance of several properties on three real-world networks: **Enron**, **Hep-th**, and **Net-trace**.

We examined five properties commonly measured and reported on network data. *Degree* is a distribution of the degrees of all vertices in the graph. *Path length* is a distribution of the lengths of the shortest paths between 500 randomly sampled pairs of vertices in the network. *Transitivity* (a.k.a. clustering coefficient) is a distribution of values where, for each vertex, we find the proportion of all possible neighbor pairs that are connected. *Network resilience* is measured by plotting the number of vertices in the largest connected component of the graph as nodes are removed in decreasing order of degree [2]. *Infectiousness* is measured by plotting the proportion of vertices infected by a hypothetical disease, which is simulated by first infecting a randomly chosen node and then transmitting the disease to each neighbor with the specified infection rate [27].

We measured each of these characteristics for the original input graph G_a and for a set of 200 output graphs sampled from $\mathcal{W}(\mathcal{G})$. We sampled uniformly from $\mathcal{W}(\mathcal{G})$ subject to the constraint that the minimum degree be one, since each input graph contains a single connected component, a fact assumed to be known by the adversary. We repeat this for each $k \in \{2, 5, 10, 20\}$ (each k produces a different $\mathcal{G}$). As a baseline, we also include a random graph of the same density as the original; this is equivalent to setting $k = |V|$. In the first five columns of Figure 8, we show results for output graphs with $k = 10$ only, although results for other values of k were qualitatively similar. We also indicate the variability of the measured values on the output graphs by either showing the 10th and 90th percentiles or showing a large number of curves, each of which corresponds to results for a single sampled output graph. Finally, in the rightmost column of Figure 8, we summarize the difference between the input and output graphs for all five measures as k varies. For degree, path lengths, and transitivity, the difference is the average value of the Kolmogorov-Smirnov statistic, which measures the maximum vertical distance between two cumulative distributions. For the last two properties (which are not distributions), the difference is measured in a similar way: it is the average maximum vertical difference between the curve corresponding to the input graph and each curve corresponding to an output graph.

The distributions of degree for the output graphs are qualitatively very similar to the input graphs for all three data sets with some exceptions. The degrees of the highest degree nodes are systematically reduced by the graph generalization. However, when compared to the random graph model ($k = |V|$), graph generalization at $k = 10$ produces far less bias.

This bias in the degree distribution appears to have relatively little effect on the distributions of path length and transitivity. The distribution of the output graphs closely resemble those of the input graph while the distributions of the random graphs are significantly distorted from those of the input graphs. This effect is particularly pronounced on the **Net-trace** and **Enron** data sets.

The effect is similar for resiliency and infectiousness. In many cases, the random graphs are significantly distorted from the input graph, while the results for the output graphs for $k = 10$ are much more similar or nearly identical to the results for the input graph. The effect is particularly pronounced for resiliency of **Net-trace**.

Finally, the summary plots show how the distortion varies with k. In general, distortion increases slowly with k, and is maximized for $k = |V|$ (the random graph baseline). For degree distribution and path length particularly, increasing values of k seems to produce only very small increases in distortion, and these values are low relative to output graphs created with the random model.

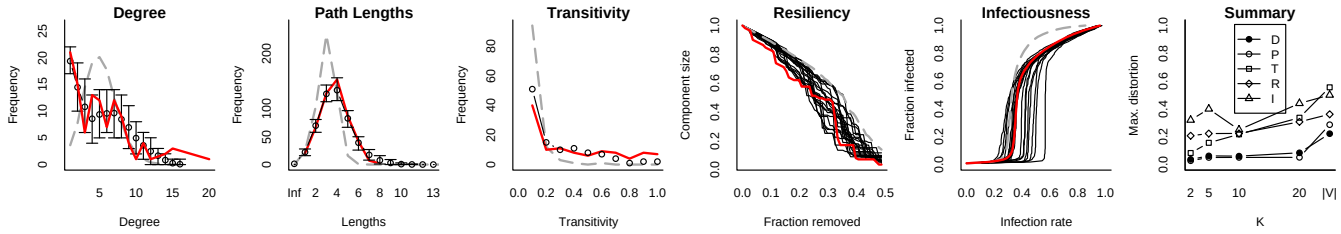
6.5 Discussion

These experimental results demonstrate that graph generalization protects privacy while allowing a range of accurate analyses. Furthermore, the proposed algorithm is an instance of a generic framework that can accommodate alternative utility objectives or privacy criteria. For a utility objective, we chose maximum likelihood, which measures the fidelity of the output graph independent of any particular analysis. The results indicate that it works well, but it is also possible to design a utility objective so that the accuracy of targeted analyses is maximized. In addition, the privacy criterion could also be changed; for instance, if the adversary knowledge was bounded by $\mathcal{H}_i$, the algorithm could allow nodes naturally hidden in large $\mathcal{H}_i$ equivalence classes to be placed in partitions smaller than size k.

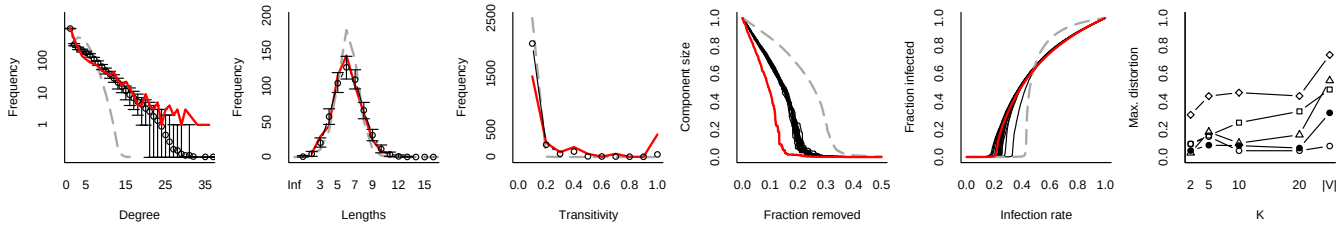
Even without changing the algorithm, the utility of the output may be improved by enhancing the analysis of the output, the generalized graph $\mathcal{G}$. For example, rather than sample graphs uniformly from $\mathcal{W}(\mathcal{G})$, the analyst could sample graphs favoring certain degree distributions. The sampling procedure could be published along with the generalized graph, without threatening privacy under our assumptions.

7. RELATED WORK

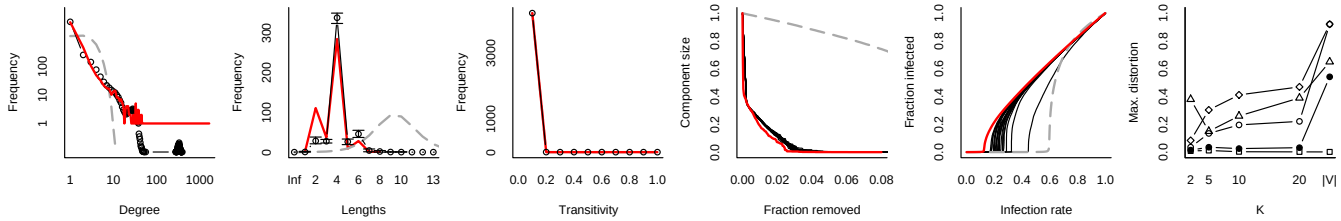
Backstrom et al. [4] were the first to propose an attack on anonymized networks, demonstrating that removing identifiers (naive anonymization) does not ensure privacy. Their main result concerns an



(a) **Enron**



(b) **Hep-th**



(c) **Net-trace**

Figure 8: Effect of anonymization on five commonly measured network properties for three different datasets. The figure compares randomly sampled anonymous graphs (black, open-circles or thin lines) to the original input graph (red, solid line) and to a random graph (gray, dashed line). The rightmost column summarizes the effect on all five measures as k varies.

active attack, where the adversary is capable of adding nodes and edges *prior* to anonymization. The attack re-identifies an arbitrary set of targets by inserting a random subgraph that will be unique with high probability, independent of the input graph, and then connecting the subgraph to the targets. We do not consider the threat of active attacks; while relevant to online social networks, they are difficult or impossible to carry out in many networks (such as email networks internal to an organization).

Passive attacks — where the adversary attacks an already published network — have been more extensively studied. This includes measures of anonymity: Singh and Zhan [24] measure the vulnerability to attack as a function of well known topological properties of the graph, and Wang et al. [26] propose a measure of anonymity based on description logic. In an earlier technical report, we introduced the models of adversary knowledge analyzed in Sections 3 and 4, and proposed the $\mathcal{H}_i$ queries as measures of re-identification risk, because of their computational efficiency and convergence (for most graphs) to automorphic equivalence [14].

In the same report, we proposed the first anonymization technique for graphs, a technique based on random edge deletions and insertions, which resisted attacks of an $\mathcal{H}_1$ adversary, but at a significant cost in graph utility [14]. Edge randomization is further explored by Ying and Wu [29], who quantify the relationship between the amount of randomization and the adversary's ability to infer the presence of an edge. They also present a randomization strategy that preserves the spectral properties of the graph; the graph utility is much improved, but the effect on privacy is not quantified.

Two groups have proposed anonymizing graphs by inserting edges until a certain level of structural uniformity is achieved. Zhou and Pei [31] consider a node-labeled graph and anonymize with respect to an adversary who knows the local neighborhood of a target (the induced subgraph of the target and its neighbors). They anonymize the graph by generalizing node labels and inserting edges until each neighborhood is isomorphic to at least $k-1$ others. Liu and Terzi [17] present an efficient graph anonymization algorithm that inserts edges into a graph until it becomes k-degree anonymous (for each node there are at least $k-1$ other nodes with the same degree). The above privacy conditions assume the adversary's knowledge is limited, and may not protect against the more powerful adversaries considered here. Furthermore, the cost of the anonymization is the number of edges added, a measure which assumes that edges have uniform cost; our approach explicitly models the likelihood of edges and penalizes the insertion of unlikely edges (as well as the deletion of likely edges). Also, anonymization by edge addition alone significantly biases the utility of the anonymized graph: degrees increase, paths become shorter, infections spread more rapidly, etc. In contrast, our algorithm outputs a distribution over graphs, which introduces relatively little bias in utility (as we show empirically) and also enables the analyst to estimate the uncertainty that the anonymization introduces by measuring variance.

Zheleva et al. [30] consider graphs with labeled edges and an adversary with a predictive model for links and knowledge of con-

straints on connections in the graph; the goal of anonymization is to prevent accurate prediction of a class of sensitive edges. The data model, threats considered, and adversary capabilities differ significantly from those treated here.

Rather than publish an anonymized network, Dwork et al. [9] propose an *interactive* mechanism, where the analyst poses queries and receives noisy answers. While the mechanism satisfies a strong notion of privacy, only queries with low "sensitivity" can be answered accurately, a condition which precludes many of the social network analyses (e.g., transitivity has $O(n)$ sensitivity). Furthermore, only a sub-linear number of queries can be answered in total (not per user). Rastogi et al. [22] present a non-interactive mechanism with equivalent privacy guarantees, but it does not address queries that require joins on the edge table, which are clearly crucial to network analysis.

Frikken and Golle [12] designed a protocol for privately assembling a graph that is distributed among a large number of parties; the output of the protocol is a naively-anonymized graph. Korolova et al. [16] consider an adversary who tries to re-assemble the graph from a set of views of local neighborhoods (obtained, for example, by breaking into user accounts of an online social network).

8. CONCLUSION

We have focused on what we believe to be one of the most basic and distinctive challenges for protecting privacy in network data sets—understanding the extent to which graph structure acts as an identifier. We have formalized classes of adversary knowledge and evaluated their impact on real networks as well as models of random graphs. We proposed anonymizing a graph by generalizing it: partitioning the nodes and summarizing the graph at the partition level. We show that a wide range of important graph analyses can be performed accurately on a generalized graph while protecting against re-identification risk.

Acknowledgments. We thank the anonymous reviewers for their insightful comments. Hay and Jensen were supported by the Air Force Research Laboratory and the Intelligence Advanced Research Projects Activity (IARPA), under agreement number FA8750-07-2-0158. Hay, Miklau, and Towsley were supported by NSF CNS 0627642; Weis by NSF CCF 0514621. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusion contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Intelligence Advanced Research Projects Activity (IARPA), or the U.S. Government.

9. REFERENCES

[1] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *STOC*, 2000.

[2] R. Albert, H. Jeong, and A.-L. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406:378, 2000.

[3] L. Babai and L. Kucera. Canonical labeling of graphs in linear average time. In *FOCS*, 1979.

[4] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou R3579X? Anonymized social networks, hidden patterns, and structural steganography. In *WWW*, 2007.

[5] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286, 1999.

[6] G. Bianconi and M. Marsili. Emergence of large cliques in random scale-free networks. *Europhysics Letters*, 74(4):740–746, 2006.

[7] W. W. Cohen. Enron email dataset, 2005.

[8] D. G. Corneil and C. C. Gotlieb. An efficient algorithm for graph isomorphism. *J. ACM*, 17(1):51–64, 1970.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[10] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.

[11] N. Friedkin. Horizons of observability and limits of informal control in organizations. *Social Forces*, 62(1):54–77, 1983.

[12] K. Frikken and P. Golle. Private social network analysis: How to assemble pieces of a graph privately. In *WPES*, 2006.

[13] J. Grochow and M. Kellis. Network motif discovery using subgraph enumeration and symmetry-breaking. In *RECOMB*, 2007.

[14] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. Technical Report 07-19, UMass Amherst, 2007.

[15] N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. In *Complexity Theory Retrospective*. Springer-Verlag, 1990.

[16] A. Korolova, R. Motwani, S. Nabar, and Y. Xu. Link privacy in social networks. In *ICDE*, 2008.

[17] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *SIGMOD*, 2008.

[18] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: privacy beyond k-anonymity. *ICDE*, 2006.

[19] D. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Halpern. Worst-case background knowledge. ICDE, 2007.

[20] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[21] J. J. Potterat, L. Phillips-Plummer, S. Q. Muth, R. B. Rothenberg, D. E. Woodhouse, T. S. Maldonado-Long, H. P. Zimmerman, and J. B. Muth. Risk network structure in the early epidemic phase of HIV transmission in Colorado Springs. *Sexually Trans. Infections*, 2002.

[22] V. Rastogi, S. Hong, and D. Suciu. The boundary between privacy and utility in data publishing. In *VLDB*, 2007.

[23] S. Russell and P. Norvig. *AI: A Modern Approach*. 2003.

[24] L. Singh and J. Zhan. Measuring topological anonymity in social networks. In *Intl. Conf. on Granular Computing*, 2007.

[25] L. Sweeney. k-anonymity: a model for protecting privacy. *Journ. of Uncertainty, Fuzziness, and KB Systems*, 2002.

[26] D.-W. Wang, C.-J. Liau, and T.-S. Hsu. Privacy protection in social network data disclosure based on granular computing. In *International Conference on Fuzzy Systems*, 2006.

[27] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[28] D. B. West. *Introduction to Graph Theory*. August 2000.

[29] X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. In *SIAM Conf. on Data Mining*, 2007.

[30] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In *PinKDD Workshop*, 2007.

[31] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *ICDE*, 2008.

APPENDIX

The appendix contains proofs of Theorems 1 and 3, which we restate here for convenience.

THEOREM 1 (SPARSE ER RANDOM GRAPHS). Let G be an ER random graph containing n nodes with edge probability given by $p = c/n$ for $c > 1$. With probability going to one, the expected sizes of the equivalence classes induced by $\mathcal{H}_i$ is $\Theta(n)$, for any $i \geq 0$.

PROOF OF THEOREM 1. Consider a network of size n. Let N_i denote the degree of the i-th node, $i \leq n$. As $n \to \infty$

$$P(N_i = k) \to \frac{c^k}{k!} e^{-c}$$

Let $M_{1,k}(n)$ denote the expected size of the equivalence class of $\mathcal{H}_1$ corresponding to node degree k when the graph is of size n and let $M_{1,k} = \lim_{n\to\infty} M_{1,k}(n)$. We have

$$\begin{aligned} M_{1,k} &= \lim_{n\to\infty} \sum_{i=1}^{n} P(N_i = k) \\ &= \Theta(n) \end{aligned}$$

A similar argument holds for $\mathcal{H}_i$, $i = 2, \ldots$. Consider a node x. We first note that the $\mathcal{H}_i$ equivalence class that x belongs to is determined by the subgraph that centered at x that includes all nodes within distance i of it. Now, as $n \to \infty$, with probability going to one, this subgraph is a tree. Moreover the probability of the above subgraph deviating from a tree is $O(1/n)$. Another observation is that every $\mathcal{H}_i$ induced equivalence class contains at least one node, whose distance i subgraph is a tree in the limit as $n \to \infty$. This follows because any $\mathcal{H}_i$ consistent multi-set can be used to construct a tree. Thus any distance i subgraph centered at a node that is not a tree is hidden by commonly found trees.

Consider a tree, t, of height i or less. Let $N(t)$ be a set containing the numbers of children for all nodes in the tree that are at distance $j = 0, 1, \ldots, i-1$ from the root. Let $G_i(x)$ denote the distance i subgraph centered at node x and let T_i denote the set of all possible height i or less trees. Then

$$\begin{aligned} P(G_i(x) = t) &= \prod_{k\in N(t)} \frac{c^k}{k!} e^{-c} + O(1/n), \quad t \in T_i \\ &= \Theta(1) \\ P(G_i(x) \notin T_i) &= O(1/n) \end{aligned}$$

Note that as n grows, the distribution of the number of children that a node within the tree has is a Poisson distribution.

Since each equivalence class contains at least one height i or less tree in the limit as $n \to \infty$, it follows from the above expressions that the expected size of each equivalence class is $\Theta(n)$. □

THEOREM 3 (POWER-LAW RANDOM GRAPHS). Let G be a PLRG on n nodes. With probability going to one, the expected sizes of the equivalence classes induced by $\mathcal{H}_i$ is $\Theta(n)$, for any $i \geq 0$.

PROOF OF THEOREM 3. The proof of Theorem 3 proceeds in a similar manner except that the Poisson distribution is replaced by $P(N_i = k) = ak^{-\alpha} > 0$, $k = 0, 1, \ldots$. Here a is chosen so that $\sum_{k=0}^{\infty} P(N_i = k) = 1$. □