Anonymization and Deanonymization of Social Network Data

Sean Chester, Bruce M. Kapron, Venkatesh Srinivasan, Gautam Srivastava, and Alex Thomo

Department of Computer Science
University of Victoria, Victoria BC Canada
PO Box 1700 STN CSC V8W 2Y2
schester@uvic.ca, bmkapron@cs.uvic.ca, venkat@cs.uvic.ca, gsrivast@uvic.ca, thomo@cs.uvic.ca

Synonyms

Social network privacy, anonymity, graph algorithms, privacy breach, complexity, adversarial knowledge

Glossary

Adversary: Somebody who, whether intentionally or not, reveals sensitive, private information
Adversarial model: Formal description of the unique characteristics of a particular adversary
Attribute disclosure: A privacy breach wherein some descriptive attribute of somebody is revealed
Identity disclosure: A privacy breach in which a presumably anonymous person is in fact identifiable
***k*-*P*-anonymity**: A condition under which any instance of *P* appears at least *k* times
Target: The particular social network member against whom an adversary is trying to breach privacy

1. Definition

As social networks grow and become increasingly pervasive, so too do the opportunities to analyze the data that arises from them. Social network data can be released for public research that can lead to breakthroughs in fields as diverse as marketing and health care. But with the release of data comes questions of privacy. *Is there any information that members of the social network would not want revealed publicly? If it is released, can somebody (an adversary) attribute that information to them?*

Anonymization is the modification of data so that sensitive information remains private. Deanonymization is the converse: re-identifying somebody in an anonymized network – or even simply learning something about them that was meant not to be attributable to them.

2. Introduction

Say we the authors wanted to stimulate research on supervisory patterns among co-authors by releasing the small social network depicted in Fig. 1. The network contains an edge between two co-authors if one supervises the other, and each vertex is labelled with the percentage that the author contributed to the research. Certainly, the labels are quite sensitive, and Alice, for example, may not want it publicly known that her contribution level was lower. To protect the privacy of the co-authors, then, the social network must first be *anonymized*. In some cases, that might be a simple enough task: just remove the names and replace them with

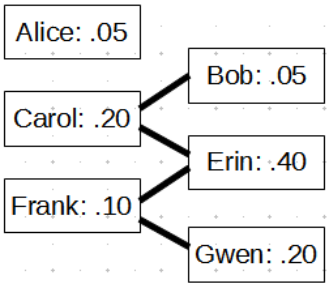

Fig. 1: Small example supervisor network. An edge (a,b) exists if a supervises b or vise versa. Vertices are also annotated with contribution percentage.

random integers.

Releasing the data makes it available for myriad analyses that the co-authors had not even anticipated. It also makes it available to Dean, an adversary who wishes to *de-anonymize* the data to uncover the sensitive information. In particular, he may want to reveal Alice's contribution level and may know that Alice, from another affiliation, has no supervisory relationship with any other author. Even after names have been stripped from the network, Dean can still exploit this background information about the *structure* of the social network graph to re-identify her and conclude her label (Backstrom, Dwork & Kleinberg 2007); viz., she is the only isolated vertex.

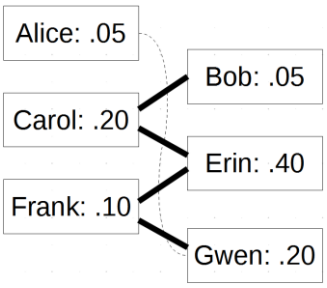

Fig. 2: A 2-anonymization of Fig. 1 by adding a fictitious edge among Alice and Gwen. Notice now that no vertex degree is unique.

Similarly, Dean may know that Erin co-supervises two of the co-authors. This is sufficient structural information to re-identify Erin, because she is the only vertex in the network connected to two other vertices who also have degree two. Something more must be done to protect the identities of Alice and Erin, and that is going to have to involve distorting the network somehow, because it is the structure of the network that reveals their identities. This is social network anonymization: distorting a social network to the point that some assumed knowledge of an adversary is rendered uninformative.

Now consider Fig. 2 in which the fictitious relationship between Alice and Gwen has been added. Now, Alice and Bob have the same degree; even if Dean knows the degree to be one, he cannot distinguish between them. Erin, too, is similarly unidentifiable, because now Frank is connected to two vertices of degree two. In fact, no matter who Dean targets, even with the knowledge of the target's neighbour's degrees, he is left with a ½ chance of guessing who is his target.

3. Key Points

Throughout this example, we knew what knowledge Dean possessed. As we continue in this chapter, we assume different levels of knowledge for Dean, and with each we ask:

- *Can* we protect ourselves from Dean's knowledge while still releasing the data?
- *When* can we do this?
- If we cannot do this, *why* not?

4. Historical Background

The need for privacy in publicly released data is not new. Relational (i.e., not social network) data has been shared for decades. Many of the ideas for social network anonymization stem from what has been researched and learned about anonymizing table data. The pivotal idea of *k*-anonymization that we introduce shortly originated with publishing relational data (Sweeney 2002). The privacy of individual table records can be well preserved if, under projection on *quasi-identifying* attributes (e.g., zip code, birthdate), the record is made identical to at least *k-1* others by a series of data suppressions.

With the onset of pervasive social networking in recent years, there has been a rush to adapt some of these ideas for social network (i.e., graph) data. The task is challenging, however, because graph structure was shown by Backstrom et al. (2007) to *quasi-identify* people itself, before even considering

the labels with which social networks are annotated. Since then, research has focused on what can, indeed, conceal one's identity (i.e., prevent *identity disclosure*) in a social network and what can conceal the attributes that describe you (i.e., prevent *attribute disclosure*).

5 Tools and Techniques for Anonymization and Deanonymization

5.1 What it means to be identical: k-anonymization formalized

In the examples from Section 2, the adversary Dean is assumed to know some local structural property *P* of his target (first, the degree of Alice and, second, the 1-neighbourhood of Erin). But by adding one edge, Alice and Erin were protected because they became structurally identical to other vertices. That is to say, the graph became *k-P-anonymous*: every vertex is identical to at least *k*-1 other vertices with respect to *P*. No matter who Dean targets with his knowledge of *P* in a *k-P*-anonymous graph, he is left with at best a 1/*k* chance of guessing the target's identity correctly.

Definition 1. *k-P-anonymous graph*. A graph G=(V,E) is *k-P-anonymous* iff the vertices can be completely partitioned into disjoint subsets such that each subset has size at least *k* and, within every subset, every vertex is identical with respect to *P*.

As a concrete example, *P* might be the *degree* of a vertex. The graph in Fig. 2 is 2-degree-anonymous. If an edge is added between Alice and Bob, the graph will become *k*-degree-anonymous for all $k \leq |V|$, since every vertex will have degree two. For a graph that is not *k-P*-anonymous, the task prior to release is to minimally distort it until it becomes *k-P*-anonymous.

Problem 1. *k-P-anonymization.* Given an input graph G=(V,E), a structural property *P*, and a privacy threshold *k*, construct a graph G'=(V,E') such that G' is *k-P*-anonymous, $E \subseteq E'$, and |E'| is minimized.

5.1.1 Anonymity with random perturbation

A first anonymization algorithm for a graph *G* is to first add *m* randomly chosen edges to produce an intermediate graph G_{int}, and then remove *m* randomly chosen edges from G_{int} to produce an anonymized graph G' (Hay et al. 2007). The choice of *m* is a balance between minimizing distortion of the graph and ensuring that $\geq k$ vertices in G' could have plausibly originated as Dean's target. By introducing randomness, Dean is forced to reason within possible world semantics and is confronted with at least *k* likely candidates as his target. So, although the resultant graph is not necessarily *k-P*-anonymous, it does leave Dean with a 1/*k* chance guessing.

5.1.2 *k*-Degree-anonymization with dynamic programming

For degree-based attacks, one can build a greedy algorithm based on the *degree sequence* of G (Lui and Terzi 2008):

Definition 2. *Degree sequence.* Given a graph G=(V,E), where the degree of a vertex v_i in V is denoted d_i, the degree sequence S_G of G is a sorted sequence of integers of length |V| wherein the frequency of any integer *i* is exactly $|\{v_j \text{ in } V: d_j=i\}|$. If the frequency of every integer is either zero or $\geq k$, the degree sequence is *k*-anonymous.

A *k*-degree-anonymous graph G will have a *k*-anonymous degree sequence. The algorithm uses dynamic programming to produce a *k*-anonymous integer sequence nearest to the degree sequence of G, then tries to produce a graph with a degree sequence matching that integer sequence. A graph can be produced from a sequence iff it meets the condition of the Erdos-Gallai Theorem for degree sequence realizability (Erdos and Gallai 1960). If the sequence does not meet that condition, then, repeatedly until success, some random noise is added to the degree sequence of G, a new sequence is

constructed, and the condition is rechecked.

From the work of Lui and Terzi (2008), the dynamic programming proceeds as follows. First, let C([1,d]) be the cost of anonymizing the first *d* integers in the sequence and let $S([a,b]) = \Sigma_{a \le i \le b}(d_b - d_i)$. Then:

For $i \le 2k$: $C([1,i]) = S([1,i])$;

For $i > 2k$: $C([1,i]) = \min \{ \min_{k \le t \le i-k} \{ C([1,t]) + S([t+1,i]) \}, S([1,i]) \}$.

If δ_i is the difference between the *i*'th integer and the largest within the same partition, then an optimal degree sequence partitioning is one which minimizes $\Sigma \delta_i$. Minimizing $C([1,|V|])$ with this dynamic programming produces an optimal partitioning. The Lui and Terzi (2008) algorithm then checks the Erdos-Gallai condition for the new sequence constructed by increasing each *i*'th integer by δ_i and, when successful, adding δ_i edges to the *i*'th vertex.

While this algorithm has no performance guarantees, experimental comparisons (Casas-Roma et al. 2012; Ying et al 2009) show that it typically reaches a *k*-degree-anonymous solution with less distortion than the random perturbation techniques. On the other hand, it is slower to reach that solution.

5.2 Broader Local Knowledge

The algorithms in Section 2 can protect a social network against an adversary Dean when Dean's knowledge is limited to the degree of his target, as he knows about Alice. But what if Dean is more powerful, as his knowledge about Erin? Several formalizations exist of a more powerful Dean, one who knows a more identifying property *P*. Correspondingly, stronger notions of *k*-*P*-anonymity are required.

5.2.1 Stronger adversarial models

To keep the examples easier to understand, we use degree and neighbourhood as the structural knowledge *P* possessed by Dean. The former leads to *k*-degree-anonymity (Lui and Terzi 2008). When Dean knows the entire neighbourhood of his target (every neighbour *and* how they are connected) (Zhou and Pei 2008), as he does with Erin, privacy requires *k*-neighbourhood-anonymity, in which the way neighbours are connected for every vertex must be identical to at least *k-1* other vertices. Many other models have been proposed. For example, Dean may know the *i-hop* neighbourhood of his target: all the neighbours within a path of length *i* (Thompson and Yao 2009). Yet stronger models have been proposed, too, based on isomorphisms (Cheng et al. 2010) and symmetry (Wu et al. 2010). While achieving each progressively stronger anonymity requirement offers greater privacy protection (presumably at the cost of graph utility), one must be careful of expecting too much, because, as shown in the next section, even reasonably modest adversarial models lead to NP-hard problems.

5.2.2 Complexity of *k*-*P*-anonymity

Interesting algorithms have been designed for many forms of anonymization or relation tables and social network graphs. These have been shown to perform quite well on real world data sets but do not have any theoretical performance guarantees. That is, there is no guarantee that these algorithms distort the input optimally in order to obtain the anonymized output. So, researchers investigated if there is an efficient algorithm, running in polynomial time, that can anonymize a given table or a graph using the minimum amount of modification required.

For table anonymization, a sequence of results showed that it is NP-hard to anonymize a table using the minimum number of suppressions required. These results were shown using reductions from known NP-hard graph optimization problems. Using a reduction from *Hypergraph Matching*, Meyerson and

Williams (2004) showed that *k*-anonymization of tables is NP-hard provided that the number of values an attribute can assume (*alphabet size*) is larger than the number of rows in the table. This result was improved by Aggarwal et al. (2005) who showed a hardness result for a ternary alphabet using a reduction from *Partition into Triangles*. Finally, Bonizzoni et al. (2009) obtained a hardness result for binary tables via a reduction from *Minimum Vertex Cover*.

The hardness results for anonymizing tables were, in turn, used to show NP-hardness results for graph anonymization such as 1-neighbourhood anonymization (Zhou and Pei 2008) in vertex-labeled graphs and label-sequence anonymization in edge-labeled graphs (Chester et al. 2012c). We now illustrate the main idea behind the reductions in these papers using the reduction of Zhou and Pei (2008). Given a binary table T with *n* rows, *l* columns and anonymity parameter *k*, build a bipartite graph G_T = (U, V, E). U is a set of *n* vertices labeled $\{r_1 , r_2 , \ldots , r_n \}$ corresponding to the rows of the table. V is a set of *k* copies of 2*l* vertices labeled $\{c_{10} , c_{11} , c_{20} , c_{21} , \ldots , c_{l0} , c_{l1} \}$ corresponding to the columns of the table. If the (i, j)-entry of T is 0, draw *k* edges from vertex r_i to the *k* vertices labeled c_{j0} . If the (i, j)-entry of T is 1, draw *k* edges from vertex r_i to the *k* vertices labeled c_{j1} . It can be shown that T can be *k*-anonymized using at most *s* entry supressions if and only if the graph G_T can be 1-neighbourhood-anonmyized using at most *ks* edge additions.

So, one can construct schemes to *k-P*-anonymize a graph, and those schemes can work well in practice and preserve the utility of the graph reasonably well. But if the objective is to construct an *optimal* anonymization – or even just one with a *fixed* level of distortion – the problem is NP-Hard.

5.2.3 Alternative formulations of *k*-anonymity

Although most research models the problem of *k-P*-anonymity as in Problem 1, a few other approaches have been suggested as well. For example, one can try to achieve *k-P*-anonymity by adding vertices as well as edges to the input graph (Chester et al. 2012b), a formulation which is not yet know to be NP-hard. Also, many social networks contain vertices that do not necessarily need to be anonymous because they do not represent typical users. Consider Twitter accounts for major sports teams and celebrities, for example. In such instances, one can potentially achieve *k*-anonymity with very minimal distortion by aiming only for *subset anonymity* (Chester et al. 2012a). A particularly recent suggestion is to output a probabilistic graph wherein the anonymity requirement is satisfied by injecting uncertainty on edges rather than just adding and removing them (Boldi et al. 2012).

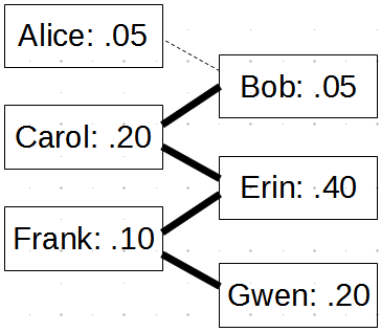

Fig. 3: An attribute-diversifying 2-degree-anonymization of Fig. 1. Now the label range for degree-1 vertices is [.05, .20].

5.2.4 Attribute disclosure

In another type of attack, the adversary Dean is not necessarily interested in identifying his target, but merely inferring her label. Such an attack is called *attribute disclosure*. Consider again the 2-degree-anonymization in Fig. 2. Despite knowing that her degree is one, Dean is unable to ascertain which vertex represents Alice and which represents Bob. He *can* infer, however, that Alice's contribution is .05, because the label is the same for both vertices.

The 2-degree-anonymization given in Fig. 3 achieves the same level of identity anonymization with the same number of additional edges as the anonymization in Fig. 2. This time, however, Dean's knowledge of Alice's degree can only reveal Alice's contribution to be within the range [.05, .20], because Alice is now in the same equivalence class as Gwen, not Bob.

The new anonymization also expands the label range for the degree-2 vertices from [.10, .40] to [.05, .40]. So, Fig. 3 offers an anonymization that better protects the sensitive information about everyone.

If Dean can infer which vertex is Alice or if Alice's equivalence class has a small label range (like in Fig. 2), then attribute disclosure will occur. So, *k*-anonymity is necessary, *but in addition to that* some attribute concealment condition must also be met. The graph is *l-diverse* (an adaptation from the similar idea in table literature (Machanavajjhala et al. 2007)) if each equivalence class contains at least *l* different labels (Zhou and Pei 2008). A graph could also be made α-*proximal* (an adaptation for graphs of *t-closeness* (Li et al. 2007)) if the distribution of labels in each vertex's neighbourhood is within α of the distribution across the entire network (Chester et al. 2012d).

With distortions to a social network that sufficiently diversify attribute labels among equivalence classes (defined by *P*) that are sufficiently large, Dean's local knowledge about graph structure can, in fact, be rendered uninformative. But what if Dean's knowledge goes beyond that?

5.3 De-anonymization beyond local knowledge

As we have discussed, a commonly used approach in anonymization of social networks is *k-P*-anonymization. Before the data is released, any sensitive information associated with individual vertices of the social network graph is suppressed and a sanitized graph that only reveals edge relationships between users is released for data mining purposes. Does this method work well in practice? There is now sufficient evidence that it *does not*. It has been shown that de-anonymization attacks can be used to extract sensitive information about certain users from such an anonymized graph by an adversary whose knowledge is *global* in nature.

Backstrom, Dwork and Kleinberg (2007) showed how active and passive attacks can be used to reveal true identities of specific users easily by an adversary whose only knowledge is an identity-anonymized version of the social network graph. An active adversary can create a small number of dummy nodes with a special edge pattern among themselves and with edges to users whose privacy it wishes to violate. Later, it easily finds this edge pattern to locate the dummy nodes in the released network and hence re-identify other users in the network. They also describe passive attacks in which a group of users can collude to discover their location in the anonymized graph using the knowledge of the edge structure among themselves. This information is in turn used to violate privacy of their immediate neighbours. It was pointed out by Narayanan and Shmatikov (2009) that this approach has some limitations in practice. For example, active attacks involving a large number of nodes may not be feasible in many real-world social networks such as a phone-call network. Furthermore, the lack of incoming edges to the dummy nodes in a directed graph could make the network operator suspect and identify an active attack.

Another notable work on de-anonymization is by Narayanan and Shmatikov (2009), who show that the nodes in a fully identity-anonymized social network graph (targets) can be identified quite effectively when the adversary has available another (auxiliary) social graph that has a significant overlap with the target graph. Their experiments with a crawled Twitter graph as a target graph, and a Flicker graph as auxiliary graph showed that the Twitter nodes could be recognized (de-anonymized) with a low error rate. The method used is based on first discovering the mappings of a small set of nodes in the auxiliary graph, the “seeds”, to corresponding nodes in the target graph. Then these mappings are propagated to other nodes in the neighborhoods of the seeds, and the propagation continues similarly to neighborhoods of the nodes discovered so far, until no more nodes can be discovered any further. The mapping exploration crucially depends on matching the degrees of the nodes in the auxiliary graph to degrees of the nodes in the target graph. Despite the success of the Narayanan’s and Shmatikov’s

method, what remains to be investigated is the amount of disruption that can be caused on its effectiveness when the target graph is degree-anonymized as opposed to only identity-anonymized.

A more recent work by Srivatsa and Hicks (2012) used a method similar in spirit to Narayanan and Shmatikov's to de-anonymize mobility traces. Location-based services that release anonymized data about location traces of various users gathered from smartphones and GPS sensor data have become very popular. They show that such mobility traces can be de-anonymized if the adversary has auxiliary information in the form of a social network involving the participating users. For example, they were able to de-anonymize bluetooth contact traces of a set of conference attendees using their DBLP co-authorship graph as auxiliary information.

5.4 Differential privacy

A rather different approach to anonymization is *differential privacy*, which does not require the release of data. Differential privacy provides a model for privacy-preserving analysis of statistical databases, which are collections of records, or datasets, which contain statistical information about individuals. It is characterized by a property of algorithms operating on the data, typically computing some statistical function (query) of the data. In particular, a randomized algorithm K is differentially private if for all datasets D, D' which are *close* (i.e., one may be obtained from the other by the deletion of exactly one record,) and all $S \subseteq \mathrm{Rng}(K)$,

$$\Pr[K(D) \in S] \leq e^{\varepsilon} \cdot \Pr[K(D') \in S].$$

This definition captures the intuitive requirement that the distribution of the output of a statistical function should not be significantly influenced by the participation of a particular individual. A natural concern here is the tradeoff between utility and privacy, in particular, whether it is possible to compute functions which are statistically useful while maintaining privacy. A natural approach to devising such functions is output perturbation, that is, the addition of some form of noise to the output of the statistical function. This must be done with care for example to avoid noise cancellation over a sequence of queries, but techniques based on the addition of Laplacian and other forms of noise have been proposed which provide differential privacy and lead to useful mechanisms for various problems in statistics (e.g. contingency table release) and learning theory. A further discussion of techniques and results in differential privacy is beyond the scope of this article; we refer the reader to the survey by Dwork (2008) for a detailed presentation.

In the setting of graphs, two versions of differential privacy are immediately apparent, namely node differential privacy and edge differential privacy. The definitions of both will follow the pattern for database privacy, differing only on the notion of what it means for two graphs to be close. Graphs G, G' are close in the edge setting if one may be obtained from the other by the deletion of exactly one edge, and in the node setting if one may be obtained from the other by the deletion of exactly one node, and its adjacent edges. Edge differential privacy is introduced by Nissim, Raskhodnikova & Smith (2007), where it is shown how to compute differentially private approximations of minimum spanning tree cost and number of triangles. In subsequent work (Hay et al. 2009; Karwa et al. 2011) refined techniques that are used to obtain further results, including differentially private approximations of the degree sequence. A recent paper (Kasiviswanathan et al. 2013) considers node differential privacy for problems including edge counting, small subgraph couting, and degree distribution.

6 Key Applications

Social network anonymization is a pre-processing step, much like data cleansing. Prior to the release of social network data, either to other parties or to the public in general, the data must be anonymized if the privacy of the network participants is to be protected. Consequently, the applications are as diverse

as the field of social network analysis. Differential privacy deserves particular note in this regard. Any differentially-private analysis task relies on an anonymity-preserving algorithm.

7 Future Directions

The field of social network anonymization and the opposing field of social network deanonymization are both quite young and rapidly expanding. Section 5.2.3 shows some ways in which the original notion of *k*-anonymity for graphs is being challenged, and assessing the merits of and extending these approaches needs to be done. Many schemes and techniques do exist, but there is still little secondary literature reviewing these. Finally, one cannot necessarily release social network data and be fully confident that nobody can attack it. Methods for preventing the global attacks described in Section 5.3 must first be developed.

8 Cross-References

Consequences of publishing real personal information in online social networks
Dark sides of social networking
Ethics of social networks and mining
Identification of the same identity across social networks and media sites
Privacy and social networks
Privacy in social networks, current and future research trends on
Statistical research in networks – looking forward
Synthetic datasets
Transforming and integrating social networks and social media data

9 References

Gagan Aggarwal, Tomás Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. (2005). "Anonymizing tables." In *Proc. ICDT*, pp.246–258.

Lars Backstrom, Cynthia Dwork, and Jon M. Kleinberg. (2007). "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography." In *Proc. WWW*, pages 181–190.

Paolo Boldi, Francesco Bonchi, Aristides Gionis, and Tamir Tassa. (2012). "Injecting uncertainty in graphs for identity obfuscation." *PVLDB*:5(11), pp.1376–1387.

Paola Bonizzoni, Gianluca Della Vedova, and Riccardo Dondi. (2009). "The *k*-Anonymity Problem Is Hard." In *Proc. FCT*, pp.26–37.

Jordi Casas-Roma, Jordi Herrera-Joancomart´ and Vicen¸Torra . (2012). "Comparing Random-Based and *k*-Anonymity-Based Algorithms for Graph Anonymization ." *Proc. MDAI* . Springer. pp.197–209.

James Cheng, Ada Wai-chee Fu, and Jia Liu. (2010). "K-isomorphism: privacy preserving network publication against structural attacks ." In *Proc. SIGMOD*, pp.459–470.

Sean Chester, Jared Gaertner, Ulrike Stege, and S. Venkatesh. (2012). "Anonymizing Subsets of Social Networks with Degree Constrained Subgraphs." In *Proc. ASONAM*, pp.418–422.

Sean Chester, Bruce M. Kapron, Ganesh Ramesh, Gautam Srivastava, Alex Thomo, and S. Venkatesh. (2012). "Why Waldo befriended the dummy? *k*-Anonymization of social networks with pseudo-nodes." *Soc. Netw. Anal. Min.*

Sean Chester, Bruce M. Kapron, Gautam Srivastava, and S. Venkatesh. (2012). “Complexity of social network anonymization.” *Soc. Netw. Anal. Min.*

Sean Chester and Gautam Srivastava. (2011). “Social Network Privacy for Attribute Disclosure Attacks.” In *Proc. ASONAM*, pp.445–449.

Cynthia Dwork. (2008). “Differential privacy: A survey of results.” In *Proc. TAMC*, pp.1–19.

Paul Erdos and Tibor Gallai. (1960). “Gráfok előírt fokszámú pontokkal.” *Matematikai Lapok*:11, pp.264–274.

Michael Hay, Chao Li, Gerome Miklau, and David Jensen. (2009). “Accurate estimation of the degree distribution of private networks.” In *Proc. ICDM* 2009, pp.169–178.

Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. (2007). “Anonymizing Social Networks.” University of Massachusetts Amherst Technical Report.

Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. (2011). “Private analysis of graph structure.” *PVLDB*, 4(11):1146–1157.

Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. (2013). “Analyzing graphs with node differential privacy.” In *Proc. TCC*, pp.457–476.

Ninghui Li and Tiancheng Li, and Venkatasubramanian, S. (2007). “*t*-closeness: Privacy beyond *k*-anonymity and *l*-diversity. ” In *Proc. ICDE*, pp.106–115.

Kun Lui and Evimaria Terzi. (2008). “Towards identity anonymization on graphs.” In *Proc. SIGMOD*, pp.93–106.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. (2007). “*L*-diversity: Privacy beyond *k*-anonymity.” *TKDD*: 1.

Adam Meyerson and Ryan Williams. (2004). “On The Complexity of Optimal K-Anonymity.” In *Proc. PODS*.

Arvind Narayanan and Vitaly Shmatikov. (2009). “De-anonymizing social networks.” In Proc. *IEEE Symposium on Security and Privacy*, pp.173–187.

Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. (2007). “Smooth sensitivity and sampling in private data analysis.” In *Proc. STOC*, pp. 75–84.

Mudhakar Srivatsa and Mike Hicks. (2012). “Deanonymizing mobility traces: using social network as a side-channel.” In *Proc. ACM Conference on Computer and Communications Security*, pp.628–637.

Latanya Sweeney. (2002). “k-Anonymity: A Model for Protecting Privacy.” *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems:* 10*(5)*, pp.557–570.

Brian Thompson and Danfeng Yao. (2009). “The union-split algorithm and cluster-based anonymization of social networks. ” In *Proc. ASIACCS*, pp.218–227.

Wentao Wu, Yanghua Xiao, Wei Wang, Zhenying He, and Zhihui Wang. (2010). “k-symmetry model for identity anonymization in social networks .” In *Proc. EDBT*, pp.111–122.

Xiaowei Ying, Kai Pan, Xintao Wu, Ling Guo . (2009). “Comparisons of Randomization and K-degree Anonymization Schemes for Privacy Preserving Social Network Publishing .” *Proc. SNA-KDD* . ACM.

Bin Zhou and Jian Pei. (2008). “Preserving Privacy in Social Networks Against Neighborhood

Attacks.” In *Proc. ICDE*, pp.506–515.