academic year 2025/2026

Suffia Azzurra

Final Project

Data Protection and Privacy

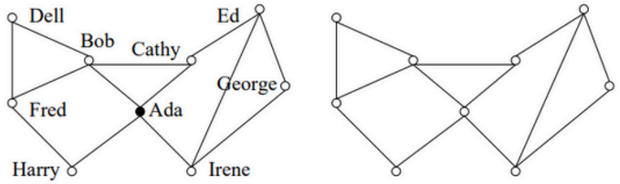
Paper

Preserving Privacy in Social Networks Against Neighborhood Attacks

Bin Zhou Jian Pei

School of Computing Science, Simon Fraser University
8888 University Drive, Burnaby, B.C., V5A1S6 Canada
{bzhou, jpei}@cs.sfu.ca

***Abstract*— Recently, as more and more social network data has been published in one way or another, preserving privacy in publishing social network data becomes an important concern. With some local knowledge about individuals in a social network, an adversary may attack the privacy of some victims easily. Unfortunately, most of the previous studies on privacy preservation can deal with relational data only, and cannot be applied to social network data. In this paper, we take an initiative**

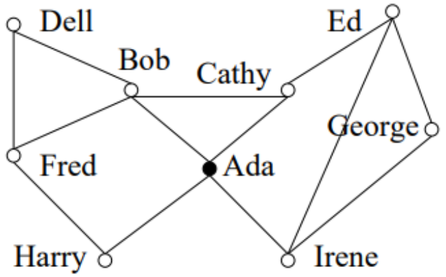


(a) the social network (b) the network with anonymous nodes

Paper Summary and Key Concepts

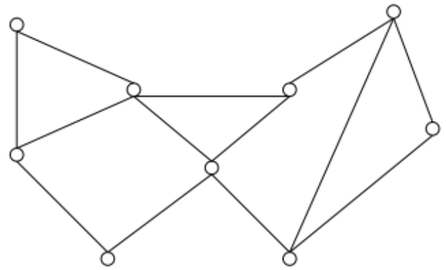
Neighborhood attack. If an adversary has some knowledge about the neighbors of a target victim and the relationship among the neighbors, the victim may be re-identified from a social network even if the victim's identity is preserved using the conventional anonymization techniques.

Data anonymizer



original graph G

conventional anonymization techniques

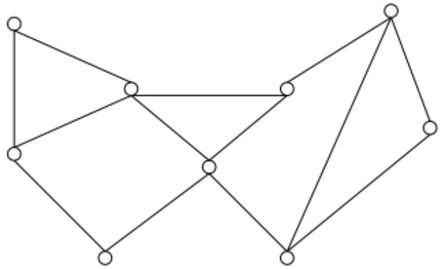


anonymized graph G'

Paper Summary and Key Concepts

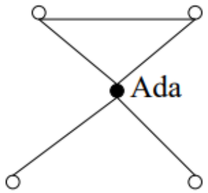
Neighborhood attack. If an adversary has some knowledge about the neighbors of a target victim and the relationship among the neighbors, the victim may be re-identified from a social network even if the victim's identity is preserved using the conventional anonymization techniques.

Data snooper



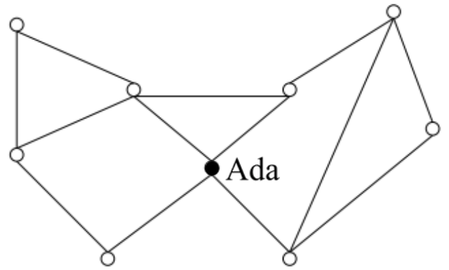
anonymized graph G'

+



background knowledge

→



identity disclosure

Paper Summary and Key Concepts

Social network: a simple graph G = (V, E, L, *L*), where:

- V is a set of vertices
- E ⊆ V × V is a set of edges
- L is a set of labels
- a labeling function *L*: V → L assigns each vertex a label

***Neighbor*$_G$(u)**: the neighborhood of u ∈ V(G) is the induced subgraph of the neighbors of u.

Consider a social network G = (V, E, L, *L*) and the anonymization G′ = (V′, E′, L′, *L*′) for publishing.
We assume that:

- **no fake vertices are added in the anonymization**, i.e. there is a bijection function A: V → V′.
- **the connections between vertices in G are retained in G′**, i.e., for (u, v) ∈ E, (A(u), A(v)) ∈ E′.

For a vertex u ∈ V (G), u is **k-anonymous** in anonymization G′ if there are at least (k − 1) other vertices $v_1, \ldots, v_{k-1} \in V(G)$ such that *Neighbor*$_{G'}$(A(u)), *Neighbor*$_{G'}$(A(v_1)), ..., *Neighbor*$_{G'}$(A(v_{k-1})) are isomorphic.

G′ is **k-anonymous** if every vertex in G is k-anonymous in G′.

Paper Summary and Key Concepts

In our social network anonymization model, there are two ways to anonymize the neighborhoods of vertices:

- *generalizing vertex labels.*
- *adding edges*.

Theorem 4 (Termination). The algorithm terminates for a finite social network of at least k vertices. In the worst case, the network will be anonymized to a clique.

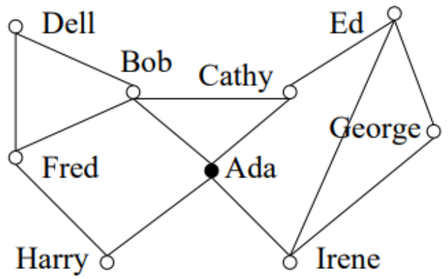
Input: a social network $G = (V, E)$, the anonymization requirement parameter k, the cost function parameters α, β and γ;
Output: an anonymized graph G';
Method:
1: initialize $G' = G$;
2: mark $v_i \in V(G)$ as "unanonymized";
3: sort $v_i \in V(G)$ as *VertexList* in neighborhood size descending order;
4: WHILE (*VertexList* $\neq \emptyset$) DO
5: let *SeedVertex* = *VertexList.head*() and remove it from *VertexList*;
6: FOR each $v_i \in$ *VertexList* DO
7: calculate *Cost*(*SeedVertex*, v_i) using the anonymization method for two vertices;
END FOR
8: IF (*VertexList.size*() $\geq 2k - 1$) DO
let *CandidateSet* contain the top $k - 1$ vertices with the smallest *Cost*;
9: ELSE
10: let *CandidateSet* contain the remaining unanonymized vertices;
11: suppose *CandidateSet*= $\{u_1, \ldots, u_m\}$, anonymize *Neighbor*(*SeedVertex*) and *Neighbor*(u_1) as discussed in Section III-B.2;
12: FOR $j = 2$ to m DO
13: anonymize *Neighbor*(u_j) and {*Neighbor*(*SeedVertex*), *Neighbor*(u_1), . . . ,*Neighbor*(u_{j-1})} as discussed in Section III-B.2, mark them as "anonymized";
14: update *VertexList*;
END FOR
END WHILE

Fig. 5. Anonymizing a Social Network.

Paper Summary and Key Concepts

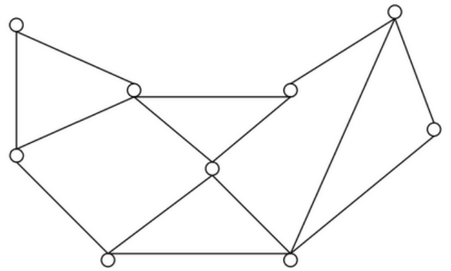
Theorem 1 (K-anonymity). Let G be a social network and G’ an anonymization of G. If G’ is k-anonymous, then with the neighborhood background knowledge, any vertex in G cannot be re-identified in G’ with confidence larger than 1/k.

Data anonymizer



original graph G

proposed anonymization method

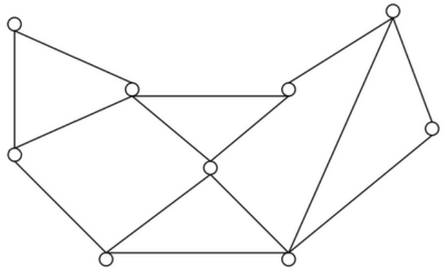


anonymized graph G’

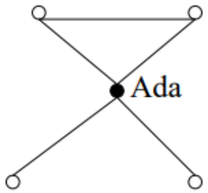
Paper Summary and Key Concepts

Theorem 1 (K-anonymity). Let G be a social network and G’ an anonymization of G. If G’ is k-anonymous, then with the neighborhood background knowledge, any vertex in G cannot be re-identified in G’ with confidence larger than 1/k.

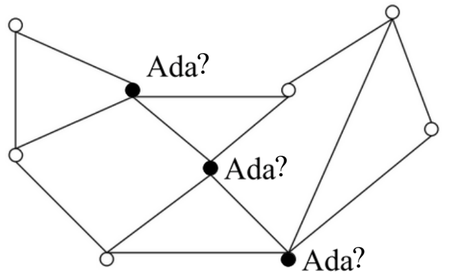
Data snooper



anonymized graph G’ +



background knowledge →



confidence ≤ 1/k

Implementation

Data generation and Implementation

Social networks are typically characterized by two main properties:

- **Property 1**: The vertex degree distribution follows a power law.
- **Property 2**: The small-world phenomenon is present.

Two graph topologies used:

- **Erdős–Rényi (ER) graphs**, commonly used as a baseline in graph analysis.
- **Barabási–Albert (BA) graphs**, model for generating synthetic networks with scale-free structure that capture the key structural properties of social networks.

No real social networks were used.

- Current implementation does not scale to large graphs within reasonable execution time.

The implementation is structured around a main component for anonymization and a additional component for graph comparison:

- **SocialAnonymizer**, which handles the anonymization process and acts as the main entry point for anonymizing graphs through the *anonymize_graph* method.
- **MDFSCoder**, which generates the minimum DFS code of a graph to obtain a canonical representation, enabling custom graph isomorphism checking.

Assumptions and Limitations

Assumptions

1. Label generalization and node labels are not considered in the anonymization strategy.
2. In *anonymize_graph*, candidate selection uses a heuristic based on degree instead of computing the full anonymization cost between two vertices.

Limitations

This implementation of the anonymization algorithm is not guaranteed to succeed in all cases.
For certain graph structures and parameter configurations, the algorithm may fail to produce a valid anonymized graph.
Failure modes include:

- Producing a **graph that is not isomorphic** with respect to the equivalence classes induced during anonymization, and
- Raising an ***AnonymizationImpossibleError*** when no suitable auxiliary node can be found outside the neighborhoods of the nodes being anonymized.

Performance

Graph isomorphism problem

Graph isomorphism problem is non-trivial (NP-intermediate), and it is invoked repeatedly.

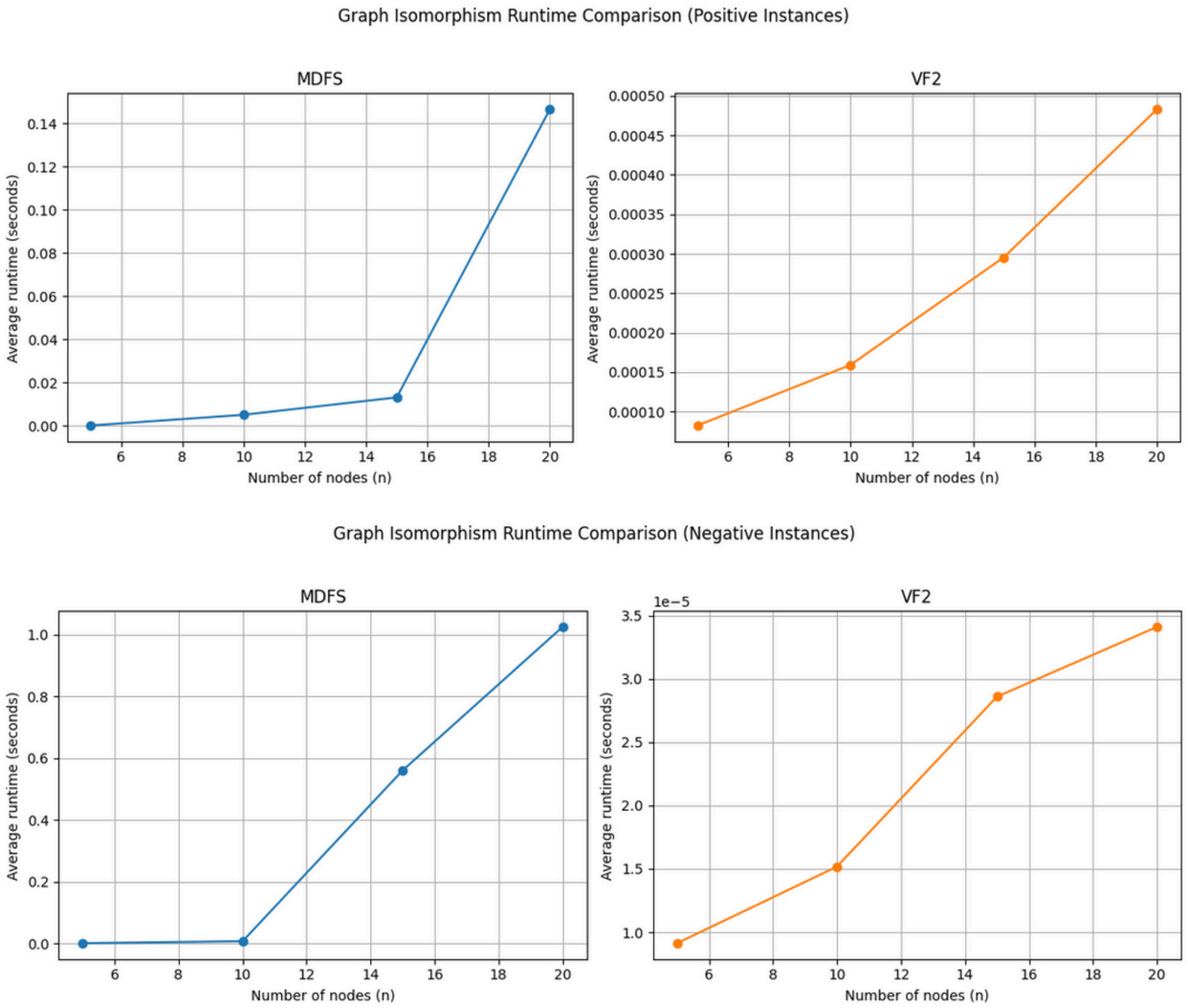
Minimum DFS Code (MDFS)

- it is a canonical graph representation.
- two graphs are isomorphic **iff** they share the same MDFS code.
- O(N!) in worst-case.
- method proposed in the paper to match isomorphic components.

VF2 algorithm

- incrementally construct a node-to-node mapping.
- O(N!) in worst-case, but faster on average.
- provided by the NetworkX library.

VF2 was adopted for speed to replace MDFS.

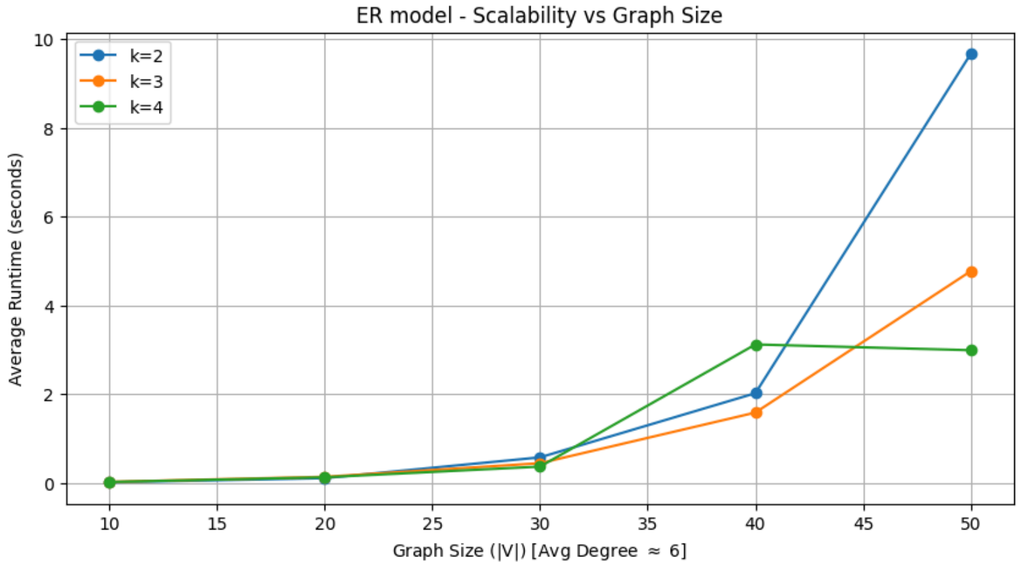


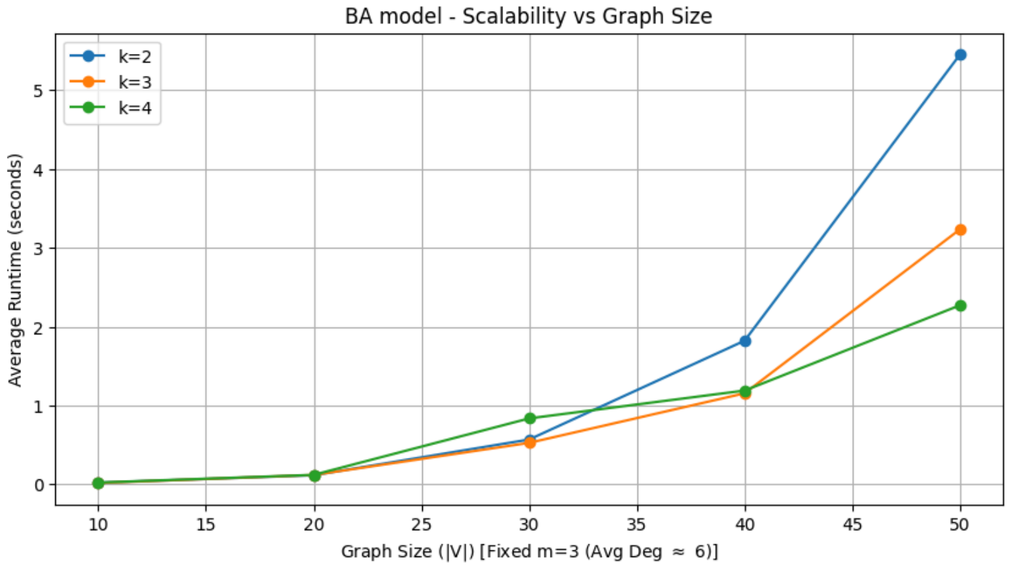
Overall algorithm performance

A formal time complexity analysis for the anonymization algorithm is not provided.
However, the time complexity of *anonymize_graph* is quite high, primarily due to:

- isomorphism checks.
- restart mechanisms.

These factors make the algorithm scale poorly as the **number of nodes** increases.





The current implementation of the algorithm does not scale to large networks in reasonable time.

Graph Metrics

Degree centrality

The degree distribution will tend to flatten. As k approaches |V|, the distribution will collapse into a single value.

Once k-anonymity is achieved:

- Each node belongs to a group of at least k vertices with isomorphic neighborhoods
- 1-neighborhood isomorphism requires the same number of adjacent vertices
- Nodes in the same group therefore share the same degree

Implication for privacy:

- No node retains a unique degree
- Degree-based re-identification attacks are effectively prevented

Betweenness centrality

As k approaches |V|, the betweenness distribution collapses to a single value, reaching zero when the graph becomes a clique (i.e., *no node acts as a bridge on shortest paths*).

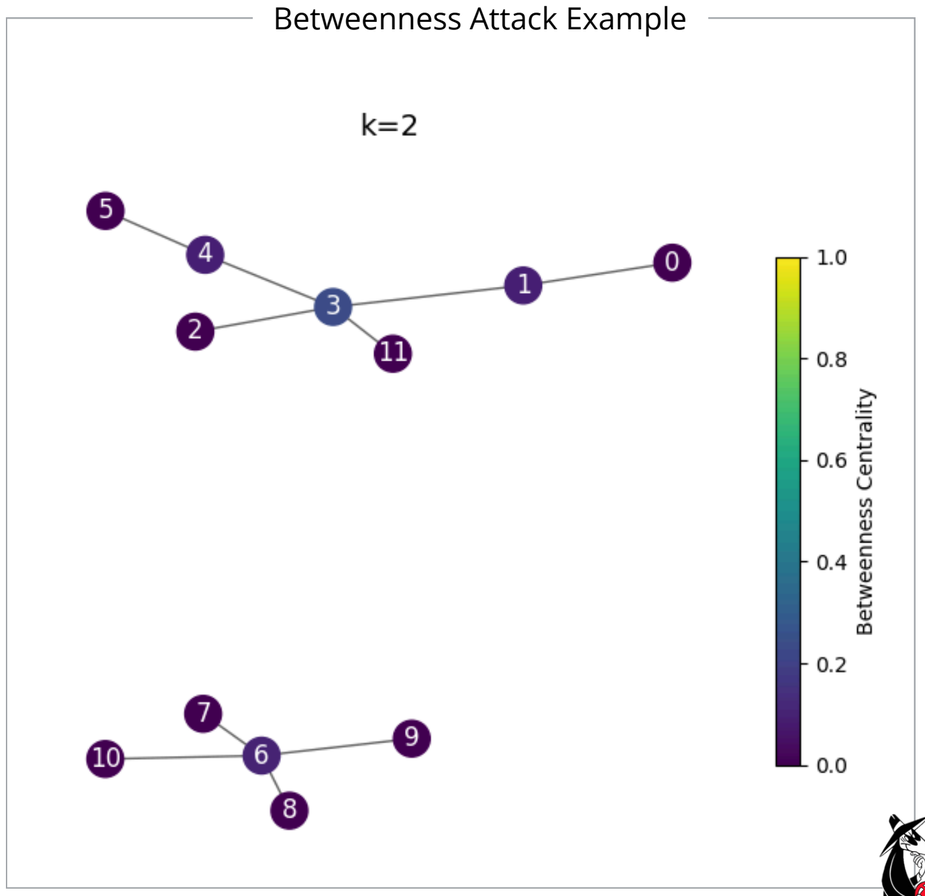
Once k-anonymity is achieved:

- Each node belongs to a group of at least k vertices with isomorphic neighborhoods
- 1-neighborhood isomorphism does not constrain betweenness centrality
- Nodes in the same group may exhibit different betweenness values

Implication for privacy:

- Nodes may retain unique betweenness centrality
- Betweenness-based re-identification attacks remain feasible

Identity disclosures exploiting betweenness



Equivalence classes:
(10, 11, 5, 7, 0, 2, 8, 9) - (1, 4) - (3, 6)

Node 3 and Node 6 have isomoprhic neighborhoods, BUT Node 3 is more central than Node 6 with respect to betweenness centrality.

Exact Value Attacks

- Node 6 betweenness: 0.1091
- Node 3 betweenness: 0.2364
- Attacker knows "*Alice has betweenness 0.2364*"

➜ Exact re-identification or negative disclosure

Ranking Attacks

- Node 3 is ranked 1, Node 6 is ranked 2 for betweenness
- Attacker knows "*Alice is in the top 8% by betweenness*"

➜ Exact re-identification or negative disclosure

Probabilistic Attacks

- Even partial knowledge of betweenness values

➜ Non-negligible re-identification probability

Closeness centrality

As k approaches |V|, the closeness distribution collapses to a single value, reaching one for normalized closeness when the graph becomes a clique (i.e., *the shortest path length between any pair of nodes is one*).

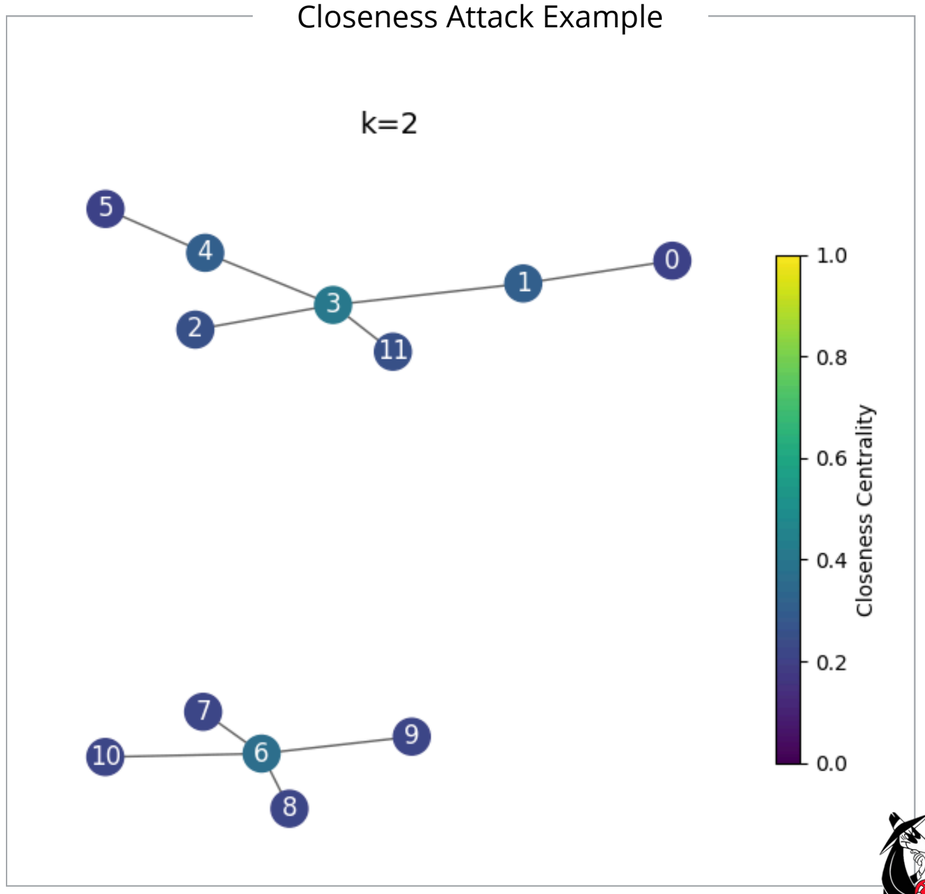
Once k-anonymity is achieved:

- Each node belongs to a group of at least k vertices with isomorphic neighborhoods
- 1-neighborhood isomorphism does not constrain closeness centrality
- Nodes in the same group may exhibit different closeness values

Implication for privacy:

- Nodes may retain unique closeness centrality
- Closeness-based re-identification attacks remain feasible

Identity disclosures expoliting closeness



Equivalence classes:
(10, 11, 5, 7, 0, 2, 8, 9) - (1, 4) - (3, 6)

Node 3 and Node 6 have isomoprhic neighborhoods, BUT Node 3 is more central than Node 6 with respect to closeness centrality.

Exact Value Attack

- Node 6 closeness: 0.3636
- Node 3 closeness: 0.40909
- Attacker knows "*Alice has closeness 0.40909*"

➜ Exact re-identification or negative disclosure

Ranking Attack

- Node 3 is ranked 1, Node 6 is ranked 2 for closeness
- Attacker knows "*Alice is in the top 8% by closeness*"

➜ Exact re-identification or negative disclosure

Probabilistic Attacks

- Even partial knowledge of closeness values

➜ Non-negligible re-identification probability

Reachability

Reachability of a node: the number of vertices reachable from that node via any path.

As k approaches |V|, reachability often increases and the number of connected components tends to decrease. However, even for k = |V|, the anonymized graph is not guaranteed to be fully connected:

- In hub-dominated graphs, the graph often collapses into a single clique
- In sparse graphs, it may converge to multiple disconnected but locally identical components

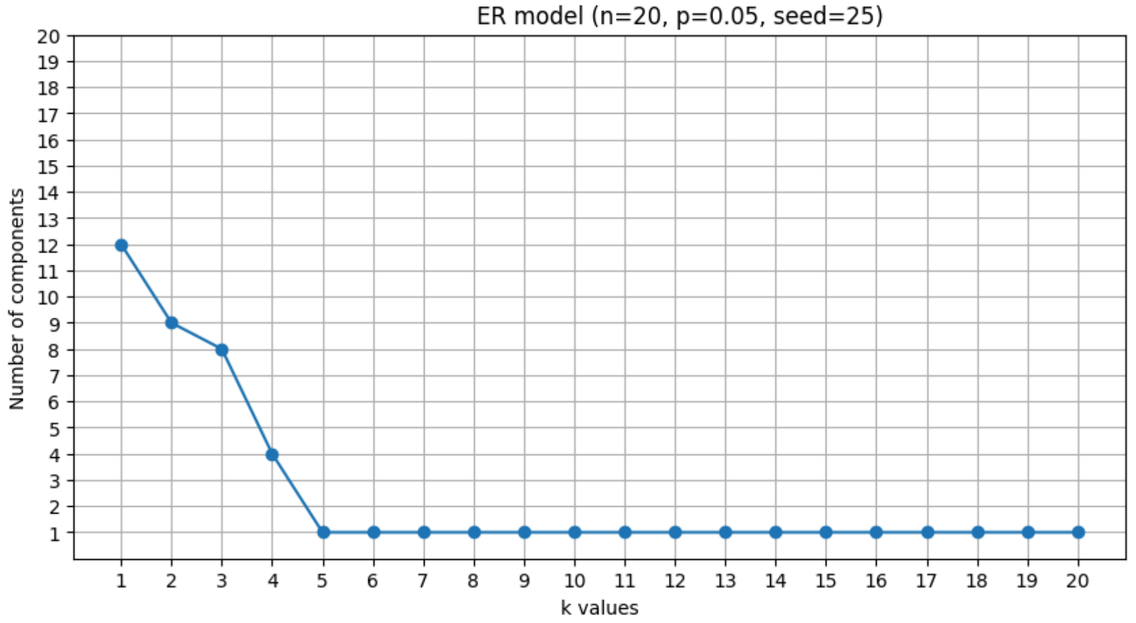
Once k-anonymity is achieved:

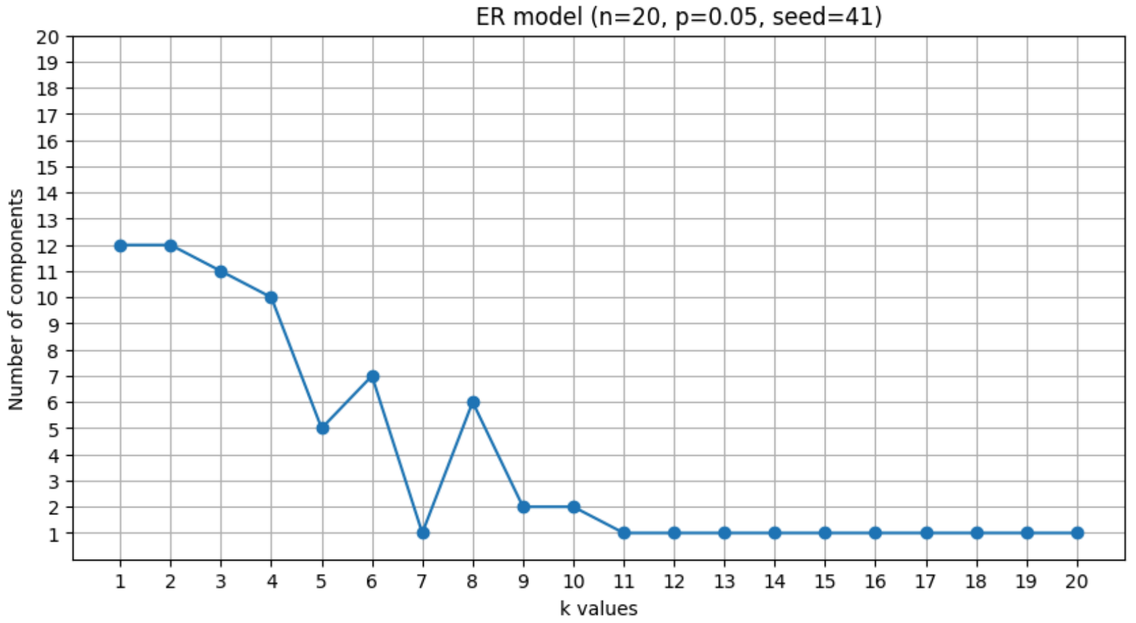
- Each node belongs to a group of at least k vertices with isomorphic 1-neighborhoods
- 1-neighborhood isomorphism does not constrain reachability
- Nodes may belong to connected components of different sizes

Implication for privacy:

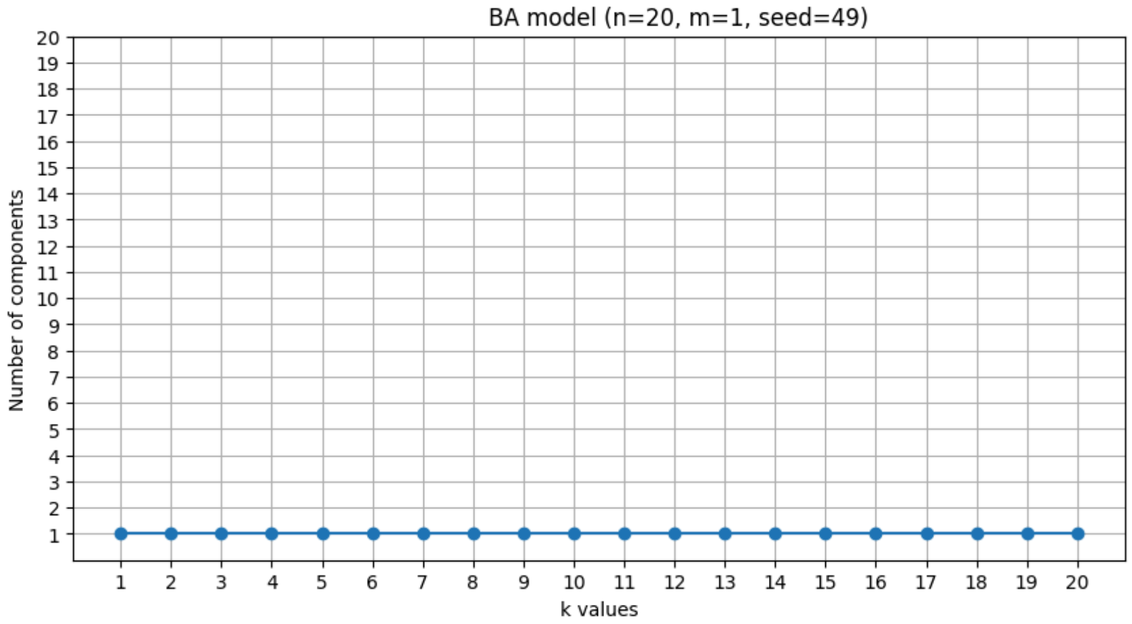
- Nodes may retain distinct reachability values
- Reachability-based re-identification remain feasible

Reachability

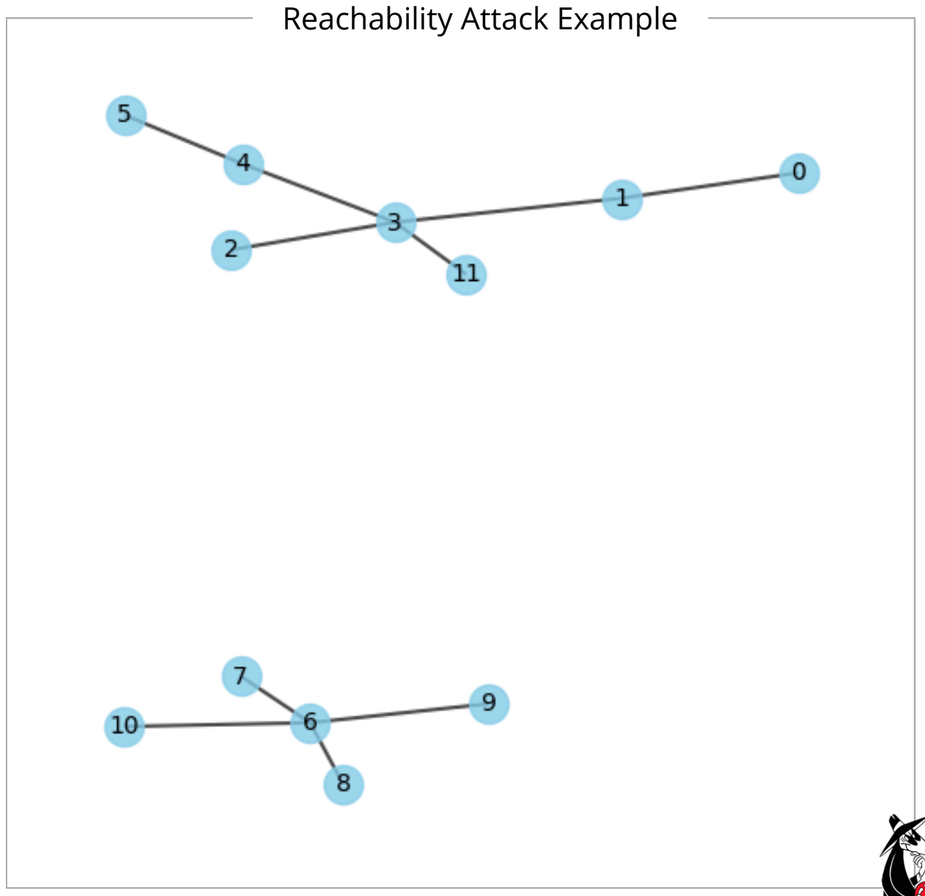




Reachability



Identity disclosures expoliting reachability



Equivalence classes:
(10, 11, 5, 7, 0, 2, 8, 9) - (1, 4) - (3, 6)

Node 3 and Node 6 have isomoprhic neighborhoods, BUT Node 3 is more central than Node 6 with respect to reachability.

Exact Value Attack

- Node 6 reachability: 5
- Node 3 reachability: 7
- Attacker knows: "*Alice can reach 7 nodes*"

➜ Exact re-identification or negative disclosure

Ranking Attack

- Node 3 is ranked 1, Node 6 is ranked 2 for reachability
- Attacker knows "*Alice belongs to the largest component*"

➜ Exact re-identification or negative disclosure

Probabilistic Attacks

- Even partial knowledge of reachability values

➜ Non-negligible re-identification probability

Summary on graph metrics

The goal of neighborhood isomorphism is to make nodes neighborhood indistinguishable ***locally*** to prevent neighborhood attacks.

Consequently:

- **Local structure attacks are effectively prevented**
  - e.g., *degree centrality*
  - these metrics become identical for all nodes in an equivalence class.
- **Global structure attacks remain a potential threat**
  - e.g., *betweenness centrality, closeness centrality, reachability*
  - local isomorphism does not completely hide a node's position within the overall network topology.

Queryability

Queryability score

The two modifications impact queryability in distinct ways:

- **Edge addition**: perturbative anonymization method, altering the topological veracity of the data.
- **Label generalization**: non-perturbative anonymization method, reducing the query space for the anonymized graphs.

We denote the queryability as the extent to which the results of queries executed on an anonymized graph resemble those obtained from the original graph. The goal is to quantify the **overall distortion** due to the anonymization process, i.e. how much the answer to queries change on average.

Query set

- Scalar queries:
  - number of edges

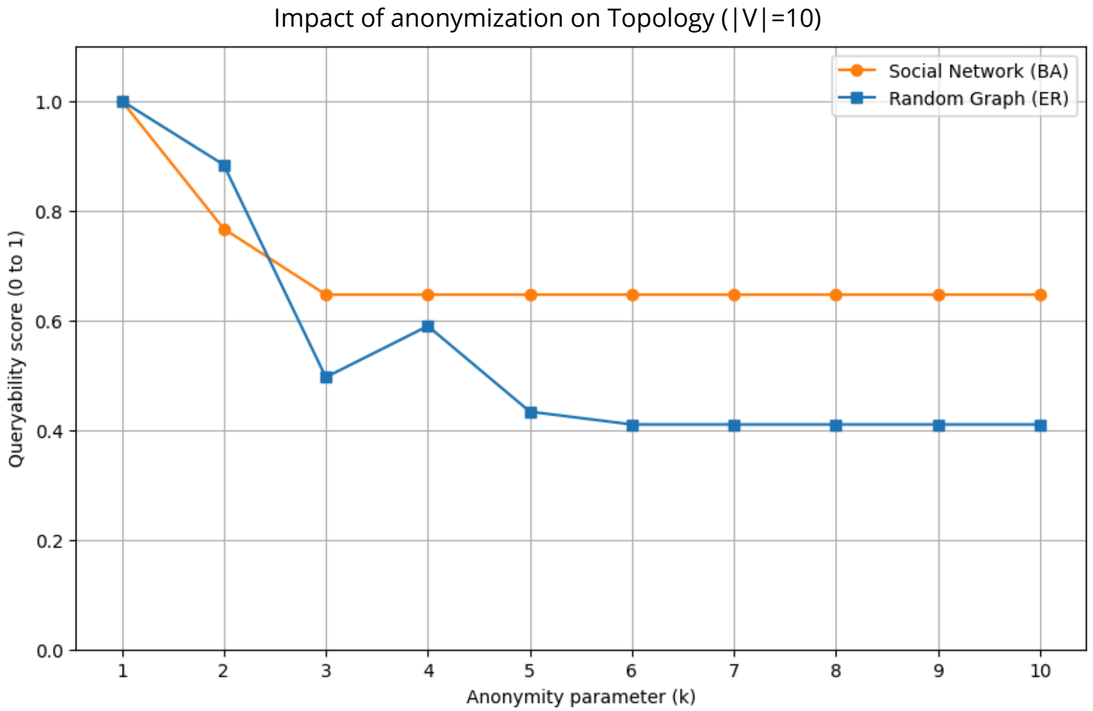
  relative error
- Vector queries:
  - degree centrality
  - betweenness centrality
  - local clustering centrality
  - closeness centrality

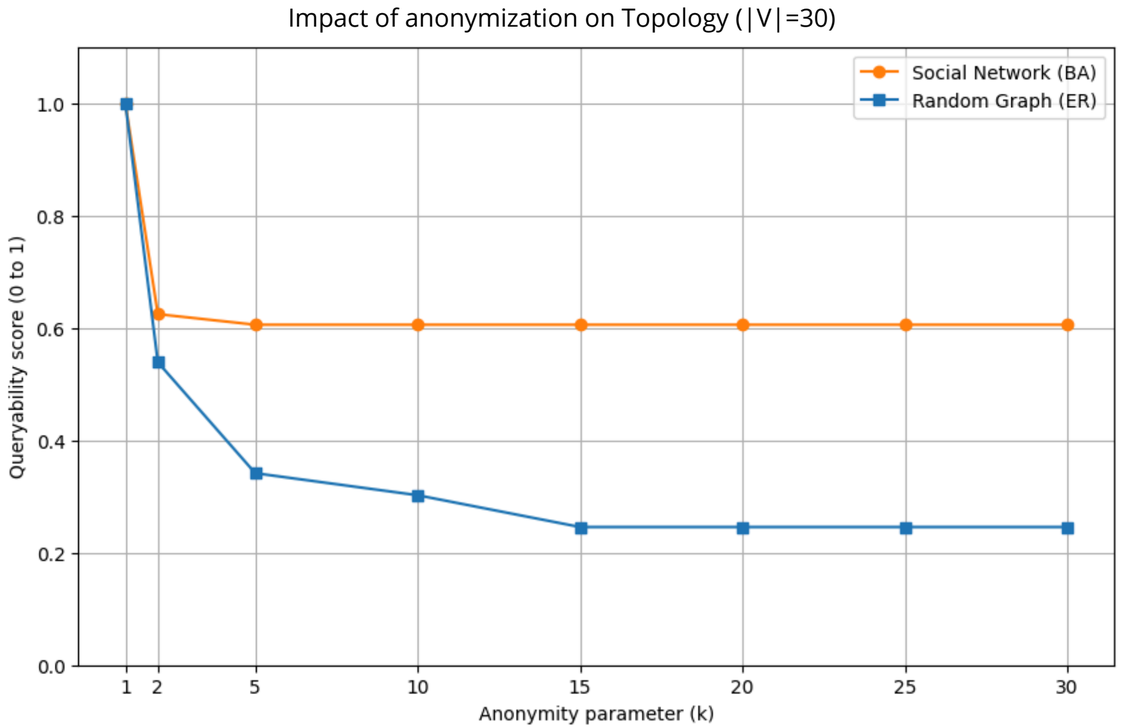
  top_k_overlap

$$Q = \frac{Q_1 + Q_2 + \ldots + Q_N}{N}$$

Queryability score

How does the anonymization impact queryability depending on the topology?





When modeling edge additions as structural perturbation:

- **Scale-free networks are known to be robust to random changes**, mantaining global properties.
- **Random graphs are more sensitive to perturbations**.

Privacy and Utility

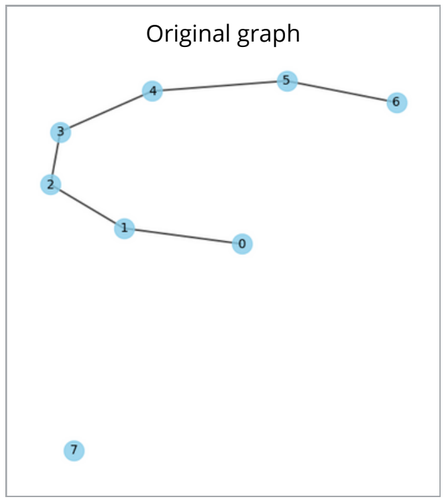
Local isomorphism

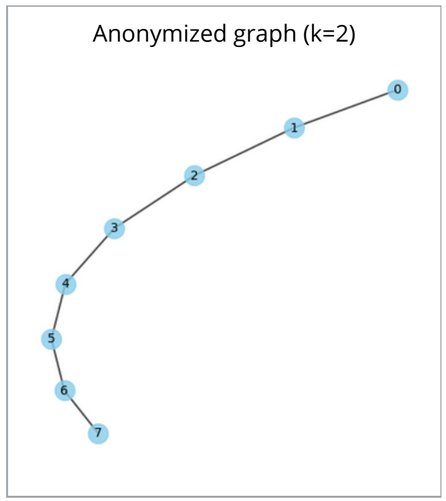
The trade-off between privacy and utility is evident:

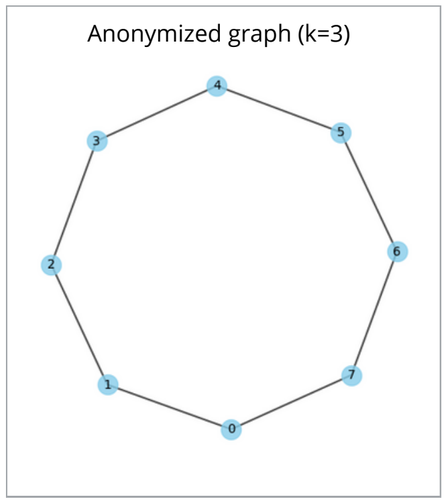
- **Highest utility, lowest privacy**: The original graph.
- **Highest privacy, lowest utility**: A clique ~~where every node label is generalized to *.~~

The anonymization algorithm is prone to forming cliques due to its construction (e.g., restart mechanism, heuristics for node mapping).
However, **enforcing k=|V| does not theoretically guarantee a clique:**

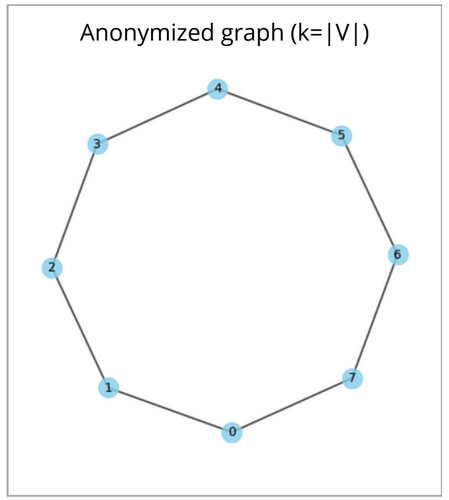
- In sparse random graphs, the result may satisfy local isomorphism without forming a clique.
- In social networks, the anonymized graph frequently converges to a clique because of hubs.







...

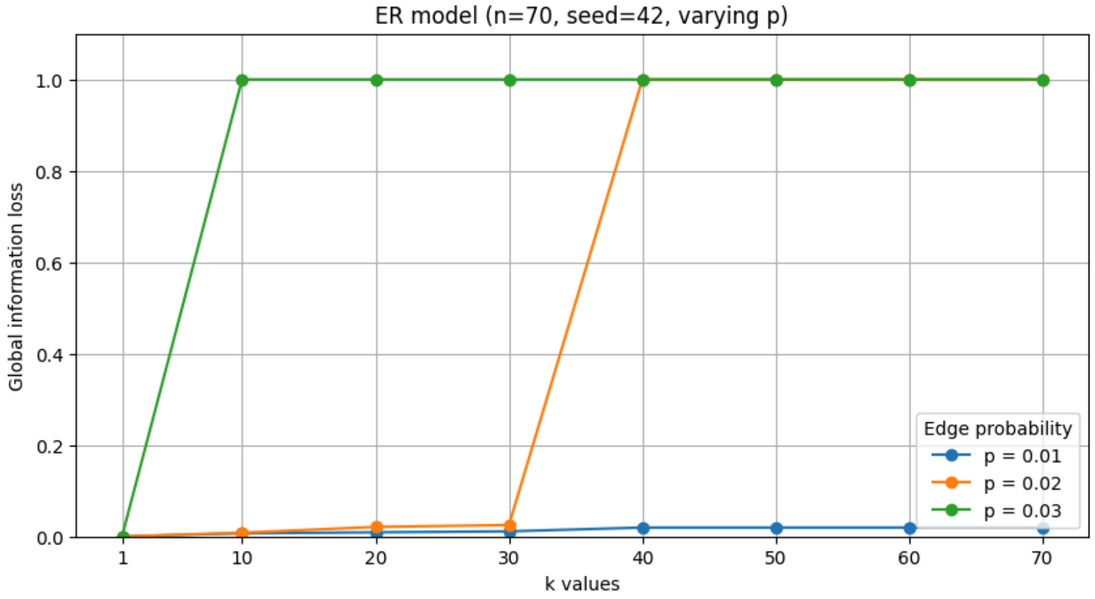


Global Information Loss

We define a **normalized global metric** representing the information loss produced by anonymization.
This metric combines the normalized number of edges added to the graph ~~and the level of label generalization~~.

When analyzing the global loss metric for increasing k:

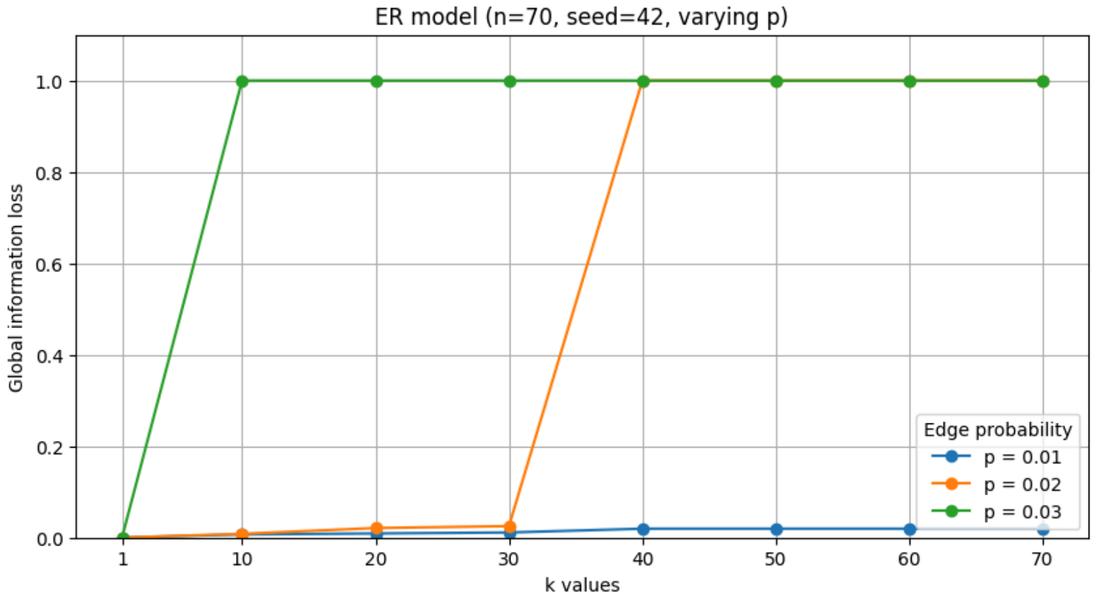
- it is common for social networks to reach the maximum loss of 1 very quickly.
- for sparse networks without hubs, the loss may converge to a value smaller than 1
  - locally isomorphic graph with fewer edges than a clique has been formed

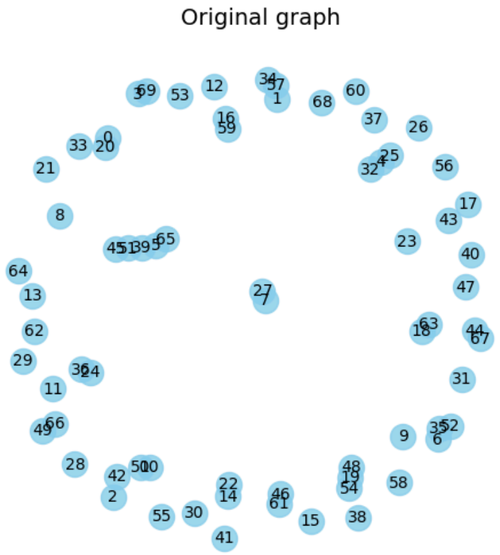


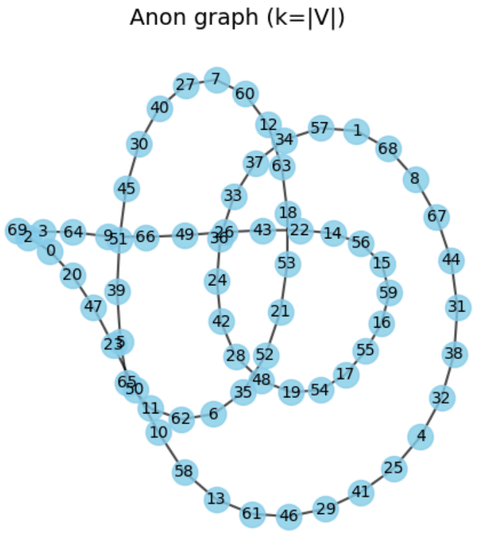
It ranges from 0 to 1 (original graph to a clique with all labels generalized to '*').

Trade-off solutions that lie between the original graph and a clique with suppressed labels are generally offered.

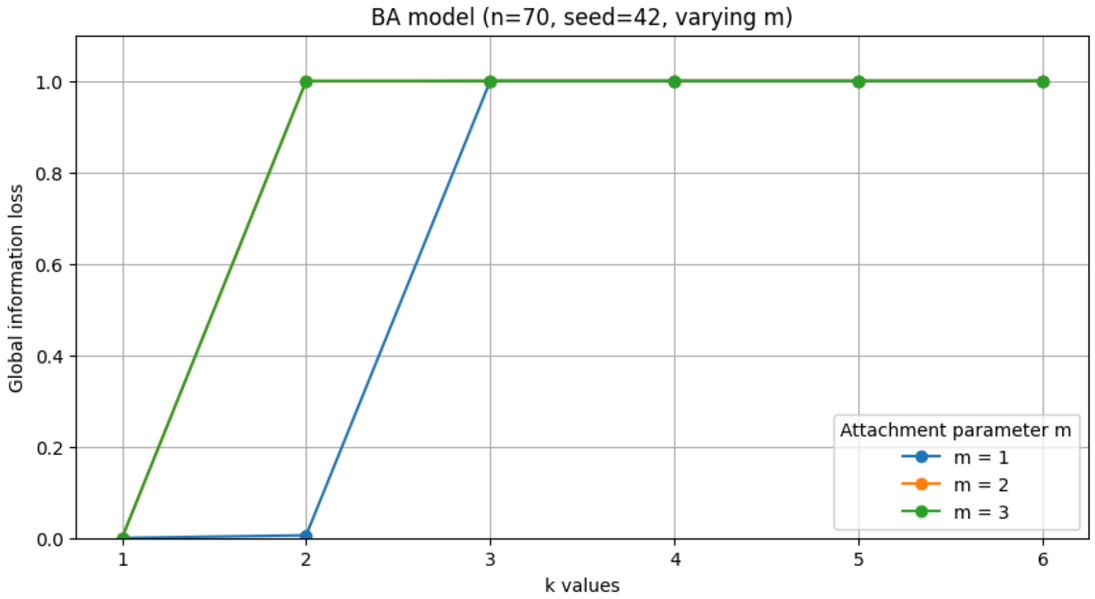
Local isomorphism example: ER graph

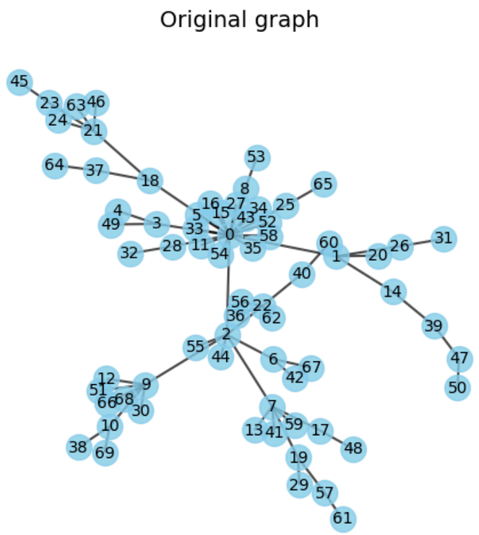


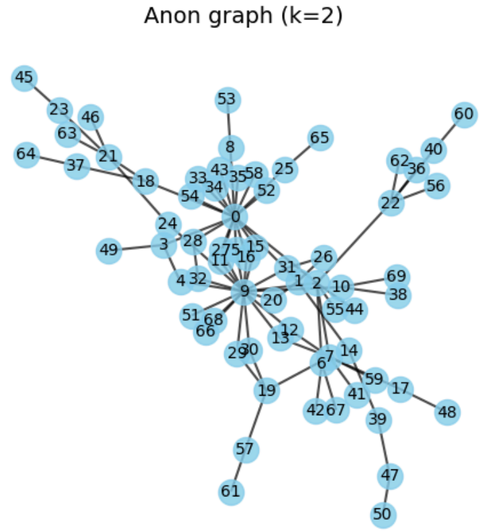




Local isomorphism example: BA graph

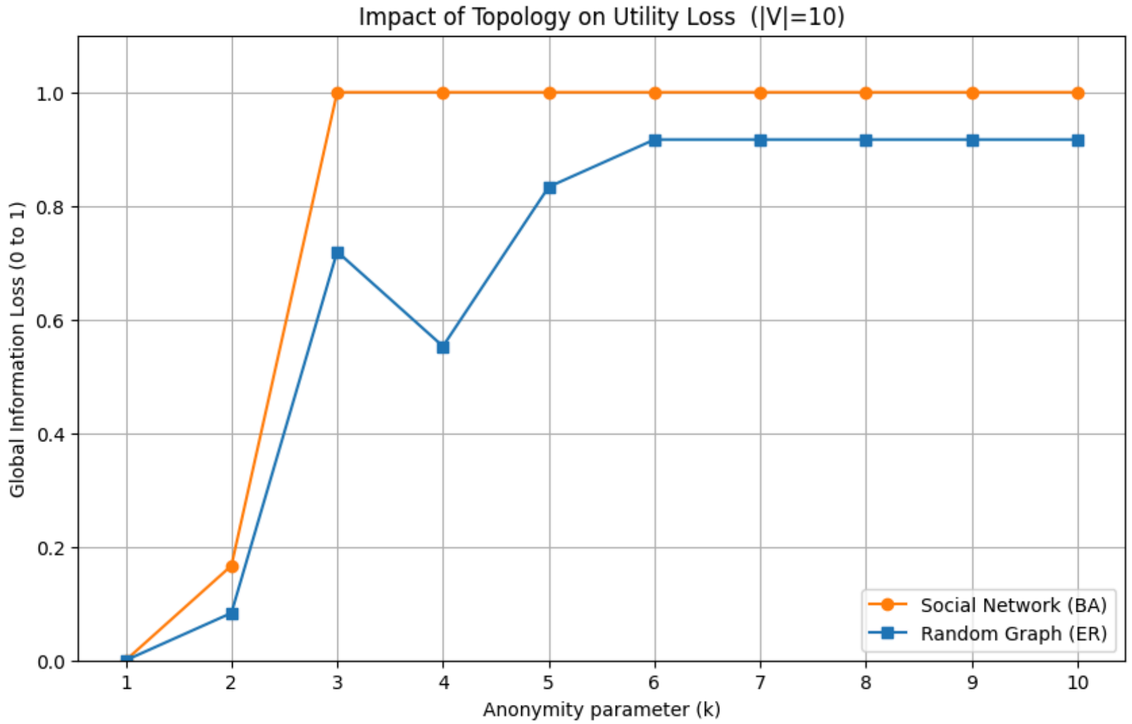


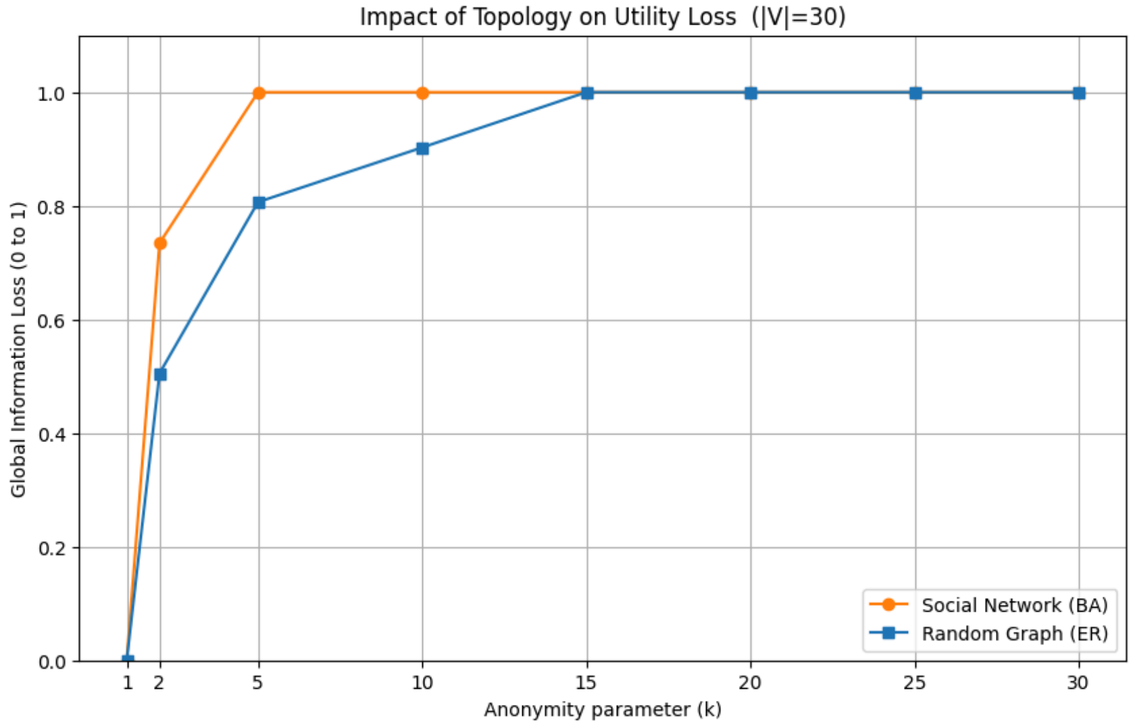




Global Information Loss and Graph Topolgy

How does the anonymization impact information loss depending on the topology?





Social network topologies exhibit a faster convergence of the global loss toward 1 (i.e., higher levels of anonymity require a rapid loss of structural information).
Social networks offer fewer intermediate trade-off solutions between privacy protection and data utility.