EURECOM[1]
Security Department
Campus SophiaTech
450 route des Chappes
CS 50193
06410 Biot Sophia Antipolis Cedex
FRANCE

Research Report RR-26-348

Exposed by Design: Topology-based privacy attacks and mitigations for Knowledge Graphs

April 22, 2026

Ibtissam Harrouche, Ferran Alborch,
Melek Önen

Tel : (+33) 4 93 00 81 00
Fax : (+33) 4 93 00 82 00
Email : {LASTNAME}@eurecom.fr

[1] EURECOM's research is partially supported by its industrial members: BMW Group, IABG, Orange, Principauté de Monaco, SAP, Norton LifeLock.

.

Abstract

Knowledge graphs (KGs) are data structures that have recently seen widespread adoption by industries for training powerful and accurate machine learning models. The advantages of KGs unfortunately come with a high cost in terms of security and privacy exposures because they increasingly use privacy-sensitive data. The common procedure to protect the confidentiality of such data by organizations is usually to publish their knowledge graphs, only partially, i.e., ensuring that privacy-sensitive information is not disclosed while the remaining subgraph becomes publicly accessible. In this paper, we show that this approach is vulnerable against privacy attacks and demonstrate that an adversary can easily infer the hidden part of the graph by simply exploiting the topology of the public KG (and nothing more). We investigate privacy attacks against KGs whose goal is to infer some private information from the graph. With this aim, we identify the most impactful features of KG topology on its leakage and design three attacks, incrementally: (1) a link inference attack to predict whether a node in the public graph exhibits a link in the private one; (2) a triple inference attack to concretely identify the hidden link with the actual tail node; and, finally, (3) a graph reconstruction attack to partially recover the hidden graph. Our experimental study shows that these attacks are successful across multiple knowledge graphs with diverse topology (up to 90% PR-AUC for link inference, 80% MRR for triple inference, and 45% graph recovery). Furthermore, to mitigate this vulnerability, we propose a topology-aware defense mechanism named Chameleon that offers KG privacy protection while guaranteeing an acceptable utility level. To address the privacy-utility trade-off, unlike existing defense strategies, Chameleon identifies the most impactful edges in the public graph, taking into account their topological features, and perturbs the graph to prevent the adversary identifies and exploit them. Through additional experiments, we show that with moderate graph perturbation budgets, Chameleon reduces attack accuracy from 92% to 50% while maintaining acceptable utility with MRR of 52% to 31% on the link prediction task. Our study, therefore, emphasizes that graph topology alone is a fundamental source of privacy leakage in KGs and that future privacy protection solutions should be topology-aware.

1 Introduction

Knowledge graphs (KGs) are increasingly gaining popularity for their ability to capture accurate information from data originating from various sources. They organize data into meaningful relationships and help machine learning models interpret complex information more easily and deliver accurate predictions. The potential of KGs has already been demonstrated across multiple domains including healthcare [30], recommendation systems [26], e-commerce [31], and knowledge-aware applications [5, 21, 22].

At a high level, knowledge graphs encode information about entities (represented as vertices on a graph) and the relation between them (encoded as edges). Different types of relation result in different "types" of edges, thus, one can interpret a knowledge graph as the accumulation of several independent graphs over the same set of vertices.

The increasing use of KGs for machine learning unfortunately comes with a high risk in terms of privacy exposures. Indeed, KGs tend to include more and more privacy-sensitive data (in healthcare applications, e.g.) which can be accidentally or maliciously disclosed to unauthorized parties. To prevent such potential breaches, the current approach consists of anonymizing the knowledge graph by modifying or removing privacy-sensitive triples before publication. For example, in a healthcare knowledge graph, a sensitive triple such as (Patient123, hasDisease, HIV) may be removed before release to avoid disclosing private medical information [15].

In this work, we investigate the actual risks of such an approach and with the design of three privacy attacks against KGs, we show that the topology of the published graph, alone, is sufficient to infer the protected, privacy-sensitive edges. More specifically:

- We propose three incremental inference attacks where an adversary exploits the topology of a graph, only: (1) a link inference attack to predict whether a node in the publicly disclosed graph exhibits an edge in the private one; (2) a triple inference attack to concretely identify the hidden edge with the actual tail node; and, finally, (3) a graph reconstruction attack to partially recover the hidden graph.

- We experimentally evaluate these attacks over three topologically distinct knowledge graphs and obtain high success: with up to 90% PR-AUC for our link inference attack, 80% MRR for triple inference attack, and lastly 45% graph recovery.

- We also study potential defense strategies and propose a novel, topology-aware, privacy solution, named Chameleon, by identifying the most impactful edges in the public KG, and perturbing the graph to prevent the adversary from exploiting them.

- We finally evaluate Chameleon as well as other potential solutions based on k-anonymity and randomized response, through various experiments on the same topologically distinct knowledge graphs and observe a significant drop of our attacks' success (from 92% to 50%) while slightly affecting their utility (in terms of link prediction MRR from 52% to 31%).

In Section 2 we expose the problem setting, going through the necessary preliminaries as well as the adversarial model, while in Section 3 we give an overview of the state of the art with respect to privacy

attacks in graphs and, more specifically, in knowledge graphs. Then, in Section 4 we introduce our framework to analyze graph topology as well as the features our attacks will use, and in Section 5 we describe our inference attacks. In Section 6 we give the experimental evaluation of our attacks, and in Section 7 we tackle the privacy defense mechanisms. We finally conclude in Section 8.

2 Background

This section establishes the formal framework for our privacy analysis. We begin with the definition of knowledge graphs and further describe the threat model.

2.1 Preliminaries

Graph Theory. A graph is defined as

$$G = (V, E)$$

where V is a set of nodes (or vertices) and E ⊆ V × V is a set of edges connecting pairs of nodes. A graph can be either non-directed, where for any two vertices v1, v2 ∈ V,

$$(v_1, v_2) \in E \iff (v_2, v_1) \in E$$

or directed, where this property does not hold and then vertices where an edge starts are called heads while vertices where an edge ends are called tails.

A neighborhood of a node v ∈ V is a subgraph defined as all other nodes v′ such that there exists a connected path to v, together with all edges connecting these nodes. More formally,

$$V_v := \{v' \in V : \exists\, v1 = v, \dots, vk = v' \in V \text{ with } (vi, vi+1) \in E \ \forall\, i \in [k-1]\}$$
$$E_v := \{(v1, v2) \in E : v1, v2 \in Vv\} \quad (1)$$

A cycle is a set of edges (v1, v2), (v2, v3), …, (vk−1, vk), (vk, v1) where vi are all distinct vertices. The length of a cycle is the number of edges (or equivalently vertices) it contains.

Knowledge Graphs. A knowledge graph is a generalization of a graph G = (V, E), where one assigns distinct semantic relations to the edges. More formally, a knowledge graph is defined as the following tuple of sets:

$$KG = (E, R, T)$$

where E is a set of vertices, R is a set of relation types, and T ⊆ E × R × E is a set of edges with specific relations in R. These edges of knowledge graphs are denoted as triples. Each triple (h, r, t) ∈ T represents a directed edge from head node h to tail node t with relation r. For each relation type r ∈ R a relation-specific subgraph

$$G_r = (E, E_r)$$

can be defined where

$$E_r = \{(h,t) \mid (h,r,t) \in T\}$$

Consequently, the knowledge graph becomes the union of these subgraphs overlaid on the shared vertex set E. This conceptualization also naturally extends the definition of neighborhood and cycles from graphs to knowledge graphs.

Public and Private Partitions. In real-world deployments, knowledge graphs are partitioned into publicly accessible and private components based on the sensitivity of the underlying information, the access control policies, or some regulatory requirements. Hence a knowledge graph can be formalized through a disjoint decomposition of the triple set:

$$KG = KG_{pub} \cup KG_{priv}$$

where the public subgraph

$$KG_{pub} = (E, R, T_{pub})$$

contains public triples, and the private subgraph

$$KG_{priv} = (E, R, T_{priv})$$

contains private triples withheld from publication. The partition is disjoint:

$$T_{pub} \cap T_{priv} = \emptyset$$

Table 1 depicts the key notation used throughout this paper.

Table 1: Notation

Notation	Description
KG = (E, R, T)	Complete knowledge graph
E	Set of nodes (all public)
R	Set of relations (all public)
T	Set of all triples
KGpub	Public subgraph (publicly released)
Tpub	Set of public triples
KGpriv	Private subgraph (not published)
Tpriv	Set of private triples
R_target	Set of target relations

2.2 Adversarial Model

We consider an adversary that has full access to the public knowledge graph

$$KG_{pub} = (E, R, T_{pub})$$

as well as to the entire set of nodes E (involved both in the private and public graphs). This representation models real-world scenarios where node identifiers or their pseudonyms are known, but the relationships between them are selectively disclosed [37]. The adversary's goal is to infer some edges in the private

graph given a set of target relation types R_target. We denote as T_target all triples (h, r_target, t) ∈ T such that r_target ∈ R_target:

$$T_{target} = \{(h, r\_target, t) \in T : r\_target \in R\_target\}$$

It is important to stress that no semantic information regarding node identities, relation labels, textual attributes, embeddings, and external knowledge sources are available to the adversary.

3 Related work

Privacy attacks on graph-based learning receive increasing attention over the years, with the motivation coming from various concerns about information leakage in machine learning models trained on sensitive graph data, such as social networks, knowledge bases, and healthcare records. While the objective of these attacks is to infer private information from trained models or published data, the various approaches differ in their threat models, adversarial capabilities, and targeted information.

Membership inference attacks aim to determine whether specific data records were included in a model's training set. Wang et al. [28] have conducted the first empirical evaluation of membership inference attacks against knowledge graph embeddings (KGE), proposing three attack types according to difficulty levels. Their experiments on medical and financial KGs demonstrate that KGE methods can leak substantial membership information. Hu et al. [16] extended these attacks to federated knowledge graph embedding (FKGE) settings, proposing novel inference attacks that exploit gradient information from distributed training. Their experiments on FB15k-237, NELL-995, and WN18RR demonstrate that federated learning does not inherently protect knowledge graph privacy, achieving attack success rates exceeding 83%. More recently, Hu et al. [17] proposed enhanced membership inference attacks with personalized differential privacy defenses on FKGE. He et al. [14] applied this to graph neural networks on general graphs, showing vulnerability even when adversaries have minimal background knowledge.

Link stealing attacks exploit trained models to infer edge existence. He et al. [13] propose the first systematic link stealing attacks on graphs, characterizing adversary knowledge along three dimensions and achieving AUC scores above 0.95 by leveraging black-box access to model predictions. Zari et al. [34] investigate link inference attacks in vertical federated graph learning on general graphs. Zhang et al. [35] study uneven vulnerability across edges in general graphs, introducing group-based attack paradigms. Wu et al. [29] extend these attacks to inductive GNN models on general graphs.

Attribute inference attacks target private node attributes by exploiting model outputs and graph properties. Gong and Liu [10, 11] demonstrate that friendship links and behavioral records in social networks enable inference of sensitive attributes. Wang et al. [27] show that aggregate group properties in general graphs can be inferred even when uncorrelated with training tasks. Zhang et al. [33] investigate how learned representations in GNNs contribute to attribute leakage on general graphs.

Graph reconstruction attacks aim to recover larger graph portions from model information. Duddu et al. [6] propose property inference, subgraph inference, and full reconstruction attacks on general graphs exploiting graph embeddings, achieving substantial attack advantages in both black-box and white-box settings. Zhang et al. [36] formalize model inversion attacks on KGs combining gradient optimization

with graph auto-encoders. Fu et al. [9] recently demonstrate near-complete topology reconstruction on general graphs through influence-based attacks, achieving high topology privacy leakage by exploiting prediction patterns.

While the vast majority of privacy attacks require access to trained models, recent work has explored reconstruction without such access. Azogagh et al. [1] propose GRAND, a method for reconstructing general undirected graphs using only common neighbor matrices derived from secure multiparty computation protocol outputs. Their work targets single-relation undirected graphs and requires access to protocol outputs, whereas our setting assumes only the published multi-relational KG is available.

This paper goes beyond existing work by investigating privacy vulnerabilities in published KGs under a realistic threat model where adversaries have no access to trained models or protocol outputs. Unlike prior attacks requiring model predictions [9, 13, 36], embeddings [16, 17, 28], gradients [39], or secure computation outputs [1], we demonstrate that the multi-relational topology of published KGs alone reveals sensitive information through novel topology-based features we identify. To the best of our knowledge, this is the first work to systematically investigate embedding-free and semantic-free privacy attacks on knowledge graphs, revealing fundamental privacy vulnerabilities that persist regardless of the protection mechanism used during publication.

4 Graph Topology

To identify the correlation between the private and public graph and therefore recover some components of the private graph, the adversary aims to exploit some topological information that first needs to be defined. More specifically, the adversary analyzes the topology around public triples containing the target relation, and then looks for similar patterns (minus the target relation) elsewhere in the graph, so as to infer the existence of target triples in the private graph.

Our first objective is therefore to identify the topological features of a graph around some target vertex v or a target triple e = (v1, r_target, v2) in a layered approach. We denote vertices at distance i from target component x as Ni(x), x being either a vertex v or a triple e = (v1, r, v2), and define the set of "depth" triples, denoted as Di(x), as triples involving vertices at distance i and i+1, and the set of "breadth" triples, denoted as Bi(x), as triples that connect two vertices that are already in the neighborhood of x at level i.

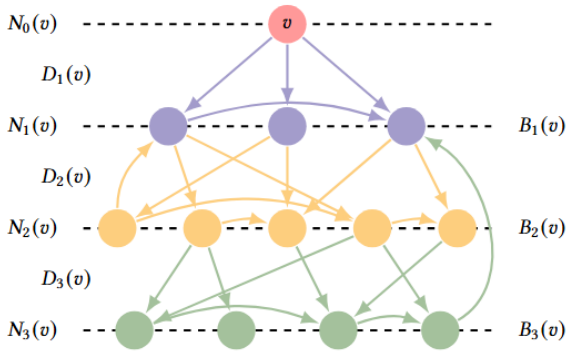


Figure 1: Neighborhood of vertex v in a graph.

More formally:

- **i-level neighbors Ni(x)** include the vertices that are at distance i from x. Recursively, it is defined as the set of all vertices neighboring Ni−1(x) that are not in Nj(x) for any j < i−1: N0(x) := {v}, if x = v, or N0(x) := {v1, v2}, if x = (v1, r, v2) given r ∈ R; and for i ≥ 1,

$$N_i(x) := \{v' \in E : \exists\, v'' \in Ni-1(x), r \in R \text{ with } (v', r, v'') \in T, \quad \text{and } v' \notin Nj(x)\ \forall\, j < i\}$$

- **i-level depth edges Di(x)** includes all triples between vertices at distance i−1 and vertices at distance i to x:

$$D_i(x) := \{(v', r, v'') \in T : v' \in Ni-1(x), \quad v'' \in Ni(x)\}$$

- **i-level breadth edges Bi(x)** includes all triples between vertices at distance i from x and other vertices at any level i ≤ j:

$$B_i(x) := \{(v', r, v'') \in T : v' \in Ni(x), \quad v'' \in Nj(x), \quad j \leq i\}$$

Note that these sets completely define the neighborhood of x (see N(v) in Equation (1) of the Background section).

> ***Remark.*** *When analyzing patterns around public vertices containing the target relation, it is especially useful to identify subgraphs around these vertices with a high density of edges, since their specific patterns are more easily recognized around other vertices where the target relation has been hidden. Note that, through this lens, breadth and depth edges contribute differently. While breadth edges inherently increase the density of the neighborhood because they do not involve any new vertices, depth edges increase density only with respect to the neighbors in the following layer.*

Despite knowledge graphs being directed, and this being key in their usefulness to train ML models, to analyze the topology around target component x and find correlation between different areas, we sometimes are interested in traversing the neighborhood of x in an *outgoing* (propagating through triples from heads to tails), *incoming* (propagating through triples from tails to heads), or *total* (propagating through triples both from heads to tails and from tails to heads) manner. As such, we update our layered definitions as follows:

- **i-level outgoing/incoming/total vertices** with respect to x are denoted as Nih(x)/Nit(x)/Ni(x), respectively, and defined as follows (with N0h(x) = N0t(x) = N0(x) as previously):

$$N_i^h(x) := \{t \in E : \exists\, h \in Ni-1h(x), r \in R \text{ with } (h, r, t) \in T, \quad \text{and } t \notin Njh(x)\ \forall\, j < i\}$$

$$N_i^t(x) := \{h \in E : \exists\, t \in Ni-1t(x), r \in R \text{ with } (h, r, t) \in T, \quad \text{and } h \notin Njt(x)\ \forall\, j < i\}$$

$$N_i(x) := \{v' \in E : \exists\, v'' \in Ni-1(x), r \in R \text{ with } (v'', r, v') \in T \text{ or } (v', r, v'') \in T, \quad \text{and } v' \notin Nj(x)\ \forall\, j < i\}$$

- **i-level outgoing/incoming/total depth triples** with respect to x are denoted as Dih(x)/Dit(x)/Di(x), respectively:

$$D_i^h(x) := \{(h, r, t) \in T : h \in Ni-1h(x), \quad t \in Nih(x)\}$$

$$D_i^t(x) := \{(h, r, t) \in T : h \in Nit(x), \quad t \in Ni-1t(x)\}$$

$$D_i(x) := \{(v', r, v'') \in T : v' \in Ni-1(x) \text{ or } Ni(x), \quad v'' \in Ni(x) \text{ or } Ni-1(x)\}$$

- **i-level outgoing/incoming/total breadth edges** with respect to x are denoted as Bih(x)/Bit(x)/Bi(x), respectively:

$$B_i^h(x) := \{(h, r, t) \in T : h \in Nih(x), \quad t \in Njh(x), \quad j \leq i\}$$

$$B_i^t(x) := \{(h, r, t) \in T : h \in Njt(x), \quad j \leq i, \quad t \in Nit(x)\}$$

$$B_i(x) := \{(v', r, v'') \in T : v' \in Ni(x), \quad v'' \in Ni(x)\}$$

Remark. *Note that the definition of total vertices, depth edges and breadth edges is equivalent to the ones that would appear when considering the graph G as non-directed. In this case (non-directed graphs), breadth edges can only exist among vertices at the same layer, thus our definition for Bi(x) (total breadth edges). For a formal proof of this claim, see the Appendix.*

We introduce one last type of notation to help in the particular case of knowledge graphs, where analyzing through the lens of distinct relations is vital. Thus, we define conditioned notation to a certain relation r (i.e. Ni|r(x) the neighbors at step i only by relation r).

While studying all previously defined features at each level (i.e. outgoing/incoming/total neighbors, breadth edges and depth edges) would give the adversary the most accurate representation of KG's topology around a target component x, this information can be a very large size given the size of KGs and consequently require powerful resources. Such a large amount of available information may not always be necessary to launch a successful attack and can even sometimes result in an unfortunate increase in false positives for the adversary's outputs. We therefore propose to make use of the following aggregate features.

- Instead of gathering the entire set of i-level vertices, the adversary may exploit their cardinality, only. This helps define the overall connectivity around component x. We therefore introduce the notion of *i-level degree* in a directed graph as follows, for * ∈ {h, t, −}:

$$n_i^*(x) := |N_i^*(x)| \tag{2}$$

- Similarly, the cardinality of the sets of i-level breadth and depth triples can be sufficient to analyze the edge density of the neighborhood, and are defined as, for * ∈ {h, t, −}:

$$b_i^*(x) := |B_i^*(x)| \quad d_i^*(x) := |D_i^*(x)| \tag{3}$$

- Finally, to exploit information related to relation types present at i-level, we define *i-level relational diversity* as, for * ∈ {h, t, −}:

$$\rho_i^*(x) := |\{ r \in R : B_{i|r}^*(x) \cup D_{i|r}^*(x) \neq \emptyset \}| \tag{4}$$

5 Topology-Based Privacy Attacks

In this section, we present three novel topology-based inference attacks that exploit topological features in the public graph KGpub to infer private triples in KGpriv. Apart from the features defined in Section 4, these attacks do not make use of any additional information such as semantic information or pretrained embeddings. The three attacks are designed in an incremental manner, gradually leaking information about the private graph.

Our starting assumption is that, due to the structured encoding of data, in real-world knowledge graphs, components that share similar links with respect to target relations, often exhibit similar topological patterns around them. For example, let us consider a medical knowledge graph with diagnosis triples linking patients to conditions. Patients sharing the same underlying medical condition such as diabetes usually have similar symptoms, treatments, medical appointments or trips to pharmacies which could be seen in the knowledge graph, which translates to similar topological patterns in their neighborhood.

With this assumption, we propose our three attacks. Given a target vertex v, the first attack's goal is to infer if there is an outgoing private edge from this vertex whereas the second one aims at identifying the actual tail vertex for that particular triple. Finally, the third attack intends to partially reconstruct the private graph.

5.1 Attack 1: Link Inference Attack

Attack 1 aims to accurately answer the following privacy question:

> *Does a target vertex htarget exhibit a link as a head vertex with relation rtarget in the private subgraph KGpriv?*

More formally, given target relation rtarget, adversary A1's goal is to determine whether there exists a triple (htarget, rtarget, t) ∈ Ttarget ∩ Tpriv without discovering the actual tail vertex t.

In KGs, vertices that are involved in the same target relation tend to share similar topological patterns in the public graph. Specifically, outgoing degree nih (Equation 2), reflects how extensively a vertex connects to others, while relation-type distribution ρih,r (Equation 4) captures the diversity of its relation types. These head-centric topological features form the basis for distinguishing participants from non-participants in a target relation. This observation leads to the following hypothesis for Attack 1:

Hypothesis 1 (Head-Centric Features for Link Inference). *The outgoing degree nih(v) and outgoing relational diversity ρih(v) of vertices in KGpub, are sufficient to accurately infer whether a target vertex is involved in a target relation.*

> ***Remark.*** *Note that this section focuses solely on inferring whether a head vertex exhibits a link with the target relation. This attack, however, can also be used to infer whether a tail vertex exhibits a link with the target relation. In this case, instead of head-centric features, the adversary would use tail-centric features in the attack.*

Attack Methodology

To identify whether vertex h exhibits a link through relation rtarget, A1 trains a binary classifier using topological features as inputs. Therefore, for each vertex h, we compute its head-centric representation φ1 as:

$$\varphi_1 : E \times KGpub \rightarrow R^{k\times 2}, \quad \varphi_1(v, KGpub) = [n_i^h(v), \quad \rho_i^h(v)]_{i\in[k]} \tag{1}$$

where k denotes the number of levels of the neighborhood analyzed.

The attack then works as follows. First, A1 builds its training dataset with the head-centric representation of public vertices exhibiting an outgoing link with respect to rtarget, and labels them as positives (D+rtarget). More formally, the positive training set is defined as follows:

$$D_+^{rtarget} = \{h \in E : \exists\, t, \quad (h, rtarget, t) \in Ttarget \cap Tpub\}$$

To define the training set of negative vertices D−rtarget, instead of randomly sampling them from the public graph, A1 selects those with connectivity comparable to positives. Degree-based hard negative sampling is used to randomly sample from all vertices with outgoing degree nih higher than the mean in D+rtarget. Degree-based selection provides computational efficiency at scale while enabling the classifier to learn fine-grained distinctions beyond simple connectivity patterns. More formally, the negative training set D−rtarget becomes a random subset of the one defined as follows:

$$D_{-,all}^{rtarget} = \{h' \in E \setminus D + rtarget : nih(h') \geq (1/|D + rtarget|)\, \Sigma h \in D + rtarget\, nih(h)\}$$

Once the training sets are defined, A1 trains binary classifier f1rtarget : Rk×2 → [0,1] on D+rtarget ∪ D−rtarget to predict whether there exists any triple (h, rtarget, *) in KGpriv. Finally, during inference, given a target head htarget ∈ E, A1 computes

$$\hat{y}(htarget;\, rtarget) = f1rtarget(\varphi 1(htarget, KGpub)),$$

where ŷ(htarget; rtarget) represents the predicted probability that htarget is involved in rtarget. Algorithm 1 formalizes the complete procedure.

Algorithm 1: Attack 1: Link Inference

Input: Public graph $\mathcal{KG}$pub = ($\mathscr{E}$, $\mathscr{R}$pub, $\mathcal{T}$pub)

Output: Inferred participation of target head htarget in rtarget within $\mathcal{KG}$priv.

1: *Training Phase*

2: $D_{+}^{rtarget}$ = {h ∈ $\mathscr{E}$: ∃t, (h, rtarget, t) ∈ Ttarget ∩ Tpub}

3: $D_{-,all}^{rtarget}$ = {h′ ∈ $\mathscr{E}$ \ $D_{+}^{rtarget}$: nih(h′) ≥ $avg_{h\in D_{+}^{rtarget}}$ nih(h)}

4: $D_{-}^{rtarget}$ ←R $D_{-,all}^{rtarget}$

5: $f_1^{rtarget} \leftarrow \text{Train}(\varphi_1, D_+^{rtarget} \cup D_-^{rtarget})$

6: *Inference Phase:*
7: for all candidate head $h \in \mathscr{E}$ do
8: $\hat{y}(h) \leftarrow f_1^{rtarget}(\varphi_1(h, \mathcal{KG}pub))$
9: end for
10: return $\{\hat{y}(h)\}_{h \in \mathscr{E}}$

5.2 Attack 2: Triple Inference Attack

Having established that link inference (i.e., Attack 1) identifies vertices that exhibit links with respect to some target relations, Attack 2 escalates the privacy leakage by formalizing the following privacy question:

> *Given that htarget exhibits a link as a head with relation rtarget in KGpriv, to whom is it connected in KGpriv?*

More formally, given a target relation rtarget and a target head htarget exhibiting the relation in KGpriv, adversary A2 aims to recover the tail vertex t such that (htarget, rtarget, t) $\in$ Ttarget $\cap$ Tpriv. Unlike link inference, which examines individual vertex properties, Attack 2 requires observing topological features of both vertices.

More in details, for (h, rtarget, t) to be a true private triple: Firstly, h should be a plausible source for rtarget (captured by head features); second, t should be a plausible target for rtarget (which can be captured by tail features); and, finally, h and t must be topologically correlated through similar neighborhoods or shared connections (which can be captured by total neighborhood features and connectivity coefficients). Accordingly, we formulate the following hypothesis for Attack 2:

Hypothesis 2 (Bidirectional topological Compatibility). *The outgoing head features (nih(h), bih(h)), incoming tail features (nit(t), bit(t)), and total neighborhood features of the triple (ni(h,t), bi(h,t), di(h,t)), together, are sufficient to accurately identify the specific target vertex in a private triple (h, rtarget, t).*

Attack Methodology

Algorithm 2 describes Attack 2. Since the public graph KGpub is assumed to be large, this attack cannot afford studying all potential pairwise vertices. To reduce the pool of potential pairs, A2 identifies the most plausible pairs using Attack 1, by finding all tails exhibiting a link with relation rtarget and further study the other relevant statistics for these pairs, only. Hence, we address accuracy and efficiency at the same time by defining a two-stage attack methodology. This, in turn, means that the structural feature extractor φ2 should contain that of Attack 1, as well as the total neighborhood features of the triple. More formally, we define the bidirectional topology representation as:

Algorithm 2: Triple Inference Attack

Input: Public graph $\mathcal{KG}$pub; threshold τh, threshold τt

Output: Inferred tails $\hat{t}$h for queried heads h

Training Phase

$f_{1,h}^{rtarget} \leftarrow$ Algorithm1.Train(φ_1^h, rtarget, $\mathcal{KG}$pub)

$f_{1,t}^{rtarget} \leftarrow$ Algorithm1.Train(φ_1^t, rtarget, $\mathcal{KG}$pub)

$D_{+,2}^{rtarget}$ = Ttarget ∩ Tpub

$D_{-,2}^{rtarget} \leftarrow$R Tpub \ Ttarget

$f_2^{rtarget} \leftarrow$ Train(ψ, $D_{+,2}^{rtarget} \cup D_{-,2}^{rtarget}$)

Inference Phase:

for all v ∈ $\mathcal{E}$ do

  ŷh(h; rtarget) ← $f_{1,h}^{rtarget}(\varphi_1^h(v, \mathcal{KG}pub))$

  ŷt(t; rtarget) ← $f_{1,t}^{rtarget}(\varphi_1^t(v, \mathcal{KG}pub))$

end for

$\mathcal{TH}^{rtarget} \leftarrow \{v \in \mathcal{E} : \hat{y}h(v; rtarget) \geq \tau h\}$

$\mathcal{TT}^{rtarget} \leftarrow \{v \in \mathcal{E} : \hat{y}t(v; rtarget) \geq \tau t\}$

for all (h,t) ∈ $\mathcal{TH}^{rtarget} \times \mathcal{TT}^{rtarget}$ do

  s(h, t; rtarget) ← $f_2^{rtarget}(\psi(h, t, \mathcal{KG}pub))$

end for

for all h ∈ $\mathcal{TH}^{rtarget}$ do

  $\mathcal{TT}h^{rtarget} \leftarrow \{t \in \mathcal{TT}^{rtarget} : (h,t) \in \mathcal{P}\}$

  $\hat{t}h \leftarrow \arg\max_{t \in \mathcal{TT}h^{rtarget}} s(h, t; rtarget)$

end for

return $\{\hat{t}h\}_{h \in Hpool}$

$$\varphi_2 : E \times E \times KGpub \rightarrow R^{k \times 7},$$
$$\varphi_2(h, t, KGpub) = [\varphi^h{}_1(h, KGpub), \varphi^t{}_1(t, KGpub), \psi(h, t, KGpub)]_{i \in [k]} \quad (2)$$

where φ1h are the head-centric features as in Equation (1), φ1t are the tail-centric features

$$\varphi^t{}_1(v, KGpub) := [n_i^t(v), \quad \rho_i^t(v)]_{i \in [k]},$$

and ψ(h, t) captures total neighborhood and connectivity features for the edge containing h and t:

$$\psi : E \times E \times KGpub \rightarrow R^{k \times 3},$$
$$\psi(h, t, KGpub) = [n_i(h, t), b_i(h, t), d_i(h, t)]_{i \in [k]} \quad (3)$$

where k denotes the number of levels of the neighborhood analyzed.

Attack 2 works as follows: First, A2 builds the training datasets for Algorithm 1 (see Attack 1's methodology), and obtains D+,1,hrtarget ∪ D−,1,hrtarget and D+,1,trtarget ∪ D−,1,trtarget using φ1h and φ1t respectively. It then builds the training set for triple classification, D+,2rtarget and D−,2rtarget, with the set of positive triples D+,2rtarget being all triples in the public graph with rtarget:

$$D_{+,2}^{rtarget} = Ttarget \cap Tpub.$$

To define the training set of negative triples D−,2rtarget, A2 samples a random subset of the remaining public triples. More formally,

$$D_{-,2}^{rtarget} \leftarrow R\ Tpub \setminus Ttarget.$$

Once the training sets are defined, following the steps of Algorithm 1, A2 trains two binary classifiers f1,hrtarget, f1,trtarget on D+,1,hrtarget ∪ D−,1,hrtarget and D+,1,trtarget ∪ D−,1,trtarget, respectively. A2 also trains a ranking model f2rtarget : Rk×3 → [0,1] on D+,2rtarget ∪ D−,2rtarget to assign a probability for a given triple (h, rtarget, t) to exist.

Finally, during inference, a two-step approach is taken:

1. Candidate Filtering: A2 executes the link inference classifiers from Attack 1 independently for heads (f1,hrtarget) and tails (f1,hrtarget) to the vertices in the public graph to obtain some plausibility scores:

$$\hat{y}_h(h;\ rtarget) = f_h^{\ rtarget}(\varphi 1(h, KGpub)),$$
$$\hat{y}_t(t;\ rtarget) = f_t^{\ rtarget}(\varphi 1(t, KGpub)).$$

Then, by applying two pre-defined thresholds τh and τt, A2 constructs target filtered pools:

$$TH^{rtarget} = \{h \in E : \hat{y}H(h;\ rtarget) \geq \tau h\},$$
$$TT^{rtarget} = \{t \in E : \hat{y}T(t;\ rtarget) \geq \tau t\}.$$

This thresholding strategy significantly reduces the search space: instead of evaluating all |E| × |E| possible pairs, A2 only evaluates |THrtarget| × |TTrtarget| pairs.

2. Pairwise Ranking: For every pair (h, t) ∈ THrtarget × TTrtarget, A2 computes

$$s(h, t;\ rtarget) = f_2^{\ rtarget}(\psi(h, t, KGpub)),$$

and finally outputs the tail with highest output of the classifier

$$\hat{t}_h = argmax_{t \in T}\ s(h, t;\ rtarget).$$

5.3 Attack 3: Graph Reconstruction Attack

Attacks 1 and 2 establish that supervised classifiers trained on topological features enable inference of individual private relationships. This raises a critical question for Attack 3:

Can those topological features enable private graph reconstruction, recovering multiple triples simultaneously rather than targeting specific relations?

Unlike Attacks 1 and 2 which target specific relations, individually, Attack 3 aims to recover multiple private triples, simultaneously, across the entire graph, spanning different relation types. The naïve approach of training and executing Attacks 1 and 2 over every possible vertex, triple and relation is not computationally feasible. We therefore propose to implement a topology-based voting method and even show that simpler methods requiring no model training can achieve effective reconstruction. While supervised classifiers achieve higher accuracy on targeted attacks (Appendix), k-NN provides the necessary efficiency for reconstructing the entire graph. This design choice once again emphasizes that KGs' privacy vulnerability strongly depends on graph topology.

Formally, if two vertices h and h′ exhibit high topological similarity (measured through topological features from Section 4) in KGpub, their private neighborhoods should be similar with high probability: KGpriv|h ≈ KGpriv|h′. Consequently, vertices with known target triples serve as templates: A3 learns correlations between public topology and private relationships from these examples, then transfers this learned pattern to topologically similar vertices for inference.

This formulates the following hypothesis:

Hypothesis 3 (Topological Analogy). *Combining head features (nih(v), bih(v)), tail features (nit(v), bit(v)), and, total features (ni(v), bi(v), di(v)) across multiple neighborhood levels are sufficient to accurately recover part of the private graph.*

Reconstruction Methodology

We formalize φ3 : E × KGpub → Rk×3 that gathers head, tail, total, and neighborhood statistics across levels L:

$$\varphi_3 : E \times KGpub \rightarrow R^{k\times 7},$$
$$\varphi_3(v, KGpub) = [n_i^h(v), b_i^h(v), n_i^t(v), b_i^t(v), n_i(v), b_i(v), d_i(v)]_{i\in[k]} \quad (4)$$

Topological similarity is measured via cosine similarity over normalized features:

$$sim(v, v') = \frac{\varphi 3(v, KGpub)\top \varphi 3(v', KGpub)}{\|\varphi 3(v, KGpub)\|2 \cdot \|\varphi 3(v', KGpub)\|2}$$

Algorithm 3 formalizes Attack 3's procedure. For each target head h ∈ E, reconstruction proceeds in three steps. First, A3 identifies the k most topologically similar vertices with known sensitive triples, forming neighborhood Uk(h). Second, for each neighbor h′ ∈ Uk(h) and its known triples (h′, rtarget, t) ∈ Ttarget, it accumulates weighted votes for each t ∈ E:

$$votes(rtarget, t) \leftarrow votes(rtarget, t) + sim(h, h')$$

Third, A3 selects candidates with scores exceeding threshold τ: Ch = {(rtarget, t) : votes(rtarget, t) ≥ τ}.

Algorithm 3 : Private Graph Reconstruction Attack

Input: Public graph $\mathcal{KG}$pub = ($\mathscr{E}$, $\mathscr{R}$pub, $\mathcal{T}$pub), threshold τ

Output: Reconstructed private graph $\mathcal{KG}$recon

$\mathcal{KG}$recon $\leftarrow \emptyset$
for all v $\in \mathscr{E}$ do
  $x_v \leftarrow \varphi_3(v, \mathcal{KG}\text{pub})$
  $\tilde{x}_v \leftarrow x_v / \|x_v\|_2$
end for
for all head h $\in \mathscr{E}$ do
  $\mathcal{U}$k(h) $\leftarrow$ TopK({h′ $\in \mathscr{E} \setminus$ {h}}, $\tilde{x}h^T \tilde{x}u$, k)
  votes(rs,t) $\leftarrow 0 \quad \forall$(rs,t) $\in \mathscr{R}s \times \mathscr{E}$
  for all h′ $\in \mathcal{U}$k(h) do
    for all rs $\in \mathscr{R}$s, t $\in \mathscr{E}$ do
      votes(rs,t) $\leftarrow$ votes(rs,t) + ($\tilde{x}h^T \tilde{x}u$)
    end for
  end for
  Ch $\leftarrow$ {(rs,t) : votes(rs,t) $\geq \tau$}
  $\mathcal{KG}$recon $\leftarrow \mathcal{KG}$recon $\cup$ {(h,rs,t) : (rs,t) $\in$ Ch}
end for
return $\mathcal{KG}$recon

6 Evaluation

Datasets. We evaluate our attacks on three knowledge graphs which are described in terms of number of vertices, relations and triples in table 2. FB15k-237 [2] consists of a general-domain graph derived from Freebase, a large-scale collaborative knowledge base. FB15k-237 is a graph with a small number of vertices (14K vertices) and high relation diversity (237 relations). NELL-995 [3], on the other hand, is an open-domain graph constructed through automatic web extraction with a much higher number of vertices (75K vertices with 200 relations). Finally, Health-KG [24] is a domain-specific biomedical graph modeling microbiome interaction which has the highest number of vertices but in exchange has much fewer relations (~1M vertices and 38 relations).

Table 2: Experimental KG statistics.

	FB15k-237	NELL-995	Health-KG
$\lvert\mathcal{E}\rvert$	14,541	75,492	950,934
$\lvert\mathcal{R}\rvert$	237	200	38
$\lvert\mathcal{T}\rvert$	310,116	154,213	6,210,399

In more topological details, Figure 2 shows the distribution of vertices' outgoing degree (nh1), relation diversity (ρh1), number of breadth edges (b1) and number of depth edges (d1). From this figure, we deduce that, FB15k-237 is a dense (compared to the two other graphs) and heterogeneous graph with high connectivity (median out-degree: 10) and high relation diversity (median: 6), indicating that vertices are involved in many relationships across diverse relation types. NELL-995 is a sparse graph with moderate heterogeneity (median out-degree: 1, relation diversity: 1). Health-KG is a sparse, homogeneous graph with minimal connectivity (median out-degree: 2) and tight relation diversity (median: 2). The experimental study evaluates the impact of these topological differences on attack success rates and defense mechanisms' performance.

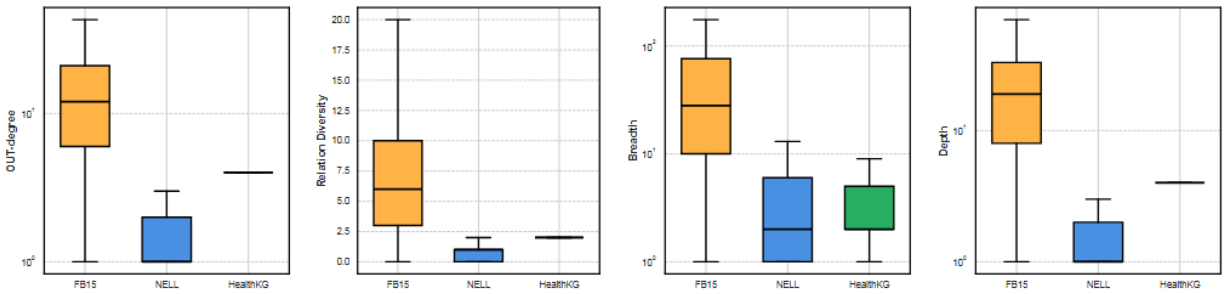


Figure 2: Topological feature distributions at level 1 neighborhood across three knowledge graphs.

Attack Implementation. Attacks 1 and 2 employ as classifier multilayer perceptrons with two fully connected layers (256, 128 units) and ReLU activations. Features are standardized using z-score normalization (zero mean, unit variance) prior to training in order to ensure that all topological features contribute equally regardless of their natural scale. Models are optimized using Adam optimizer with learning rate 10^{-3} and L2 regularization 10^{-4} for up to 200 epochs under binary cross-entropy loss. We use 20% of target relation triples as training data for the supervised models, with the remaining 80% that simulate the private knowledge graph KGpriv, reserved for testing. Attack 3 uses k = 120 nearest neighbors and cosine similarity over normalized feature vectors.

Evaluation Metrics. Attack 1 (Link Inference) is evaluated using Precision-Recall Area Under Curve (PR-AUC), a standard metric for imbalanced classification [4]. Attack 2 (Triple Inference) is evaluated using Mean Reciprocal Rank (MRR) and Hits@K which are also considered as standard for ranking tasks in KGs [2]. Attack 3 (Graph Reconstruction) combines PR-AUC to assess triple inference, along with MRR and Hits@K to evaluate the ranking quality of the recovered triples. Details on metric definitions are provided in Appendix B.

6.2 Attack 1: Link Inference Performance

Attack 1 aims to determine if head vertex h_target is involved in target relation r_target in KGpriv. A1 trains a binary classifier using topological features φ1(h, KGpub) extracted from the public graph (see Equation 6), which captures the outgoing degree and the relation-type distribution.

Table 3 presents the performance (in PR-AUC) of Attack 1 across the three knowledge graphs described in the previous section. When A1 leverages the combined feature set φ1 defined in Equation 6, the attack achieves high performance across all graphs, exceeding 90% PR-AUC for all the three of them. This confirms Hypothesis 1, i.e., that head-centric topological features are sufficient for a successful link inference. Furthermore, we have also implemented the cases where A1 only exploits one topological feature. In this case, we observe that with each feature alone, the attack exhibits significantly lower performance, demonstrating that φ1 correctly identifies the minimum necessary feature set for optimal attack performance.

Second, we investigate whether extending to k-level neighborhoods improves attack performance. In Table 3, we show that when A1 exploits level 2 features, results show marginal gains across all datasets. This demonstrates that immediate neighborhood (level 1) is already sufficient to infer links and A1 does not need to waste computational resources to go further.

From this study, we observe that FB15K-237 is the most vulnerable graph against Attack 1. This is due to the fact that the heterogeneity of this KG implies high variance in out-degree and relation diversity (which define φ1) and therefore increases the attack's success. In table 3, we also notice that the success decreases proportionally with heterogeneity.

Table 3: Link Inference (Attack 1) Evaluation (in PR-AUC) across different feature inputs

Dataset	Layer 1						Layer 2					
	n_i^h	n_i^t	b_i^h	d_i^h	ρ_i^h	ϕ_1	n_i^h	n_i^t	b_i^h	d_i^h	ρ_i^h	ϕ_1
FB15k-237	0.9213	0.0421	0.2291	0.0080	0.0329	**0.9420**	0.9580	0.3860	0.6826	0.8206	0.1162	**0.9467**
NELL-995	0.7177	0.0460	0.4106	0.0470	0.6081	**0.9249**	0.7768	0.1117	0.5424	0.1512	0.6234	**0.9109**
Health-KG	0.3081	0.2000	0.1918	0.0228	0.5933	**0.9097**	0.3160	0.3225	0.1919	0.1806	0.7449	**0.9346**

6.3 Attack 2: Triple Inference Performance

As a reminder, Attack 2 identifies the specific target vertex t such that (h_target, r_target, t) exists in KGpriv. A2 computes compatibility scores between h_target and all candidate tails obtained from Attack 1, using bidirectional topological features ψ(h, t, KGpub) (Equation 8). Candidates are ranked by these scores.

Table 4 shows the performance of Attack 2 in terms of MRR. We observe that, by exploiting φ2 (Equation 7), A2 achieves strong ranking performance across all the three knowledge graphs, confirming hypothesis 2, i.e., that bidirectional features are sufficient for successful and effective triple inference. More details on Attack 2 can be found in Appendix D (see Table 7 for Hits@3 results).

Similar to the analysis of Attack 1, we study the performance of Attack 2 when A2 exploits further levels (level 2, in this experiment). Once again, this additional knowledge does not seem to offer substantial gains and hence, immediate neighborhood features are sufficient for triple inference.

We now study the success of Attack 2 with respect to graph topology characteristics. Similar to our study in Attack 1, from table 4, we notice that FB15K-237 is, once again, the most vulnerable graph among the three. The high variance in number of breadth and depth edges shown in Figure 2 (which build ψ) justify this success. We also observe that the success rate, again, decreases proportionally with heterogeneity.

Table 4: Triple Inference (Attack 2) Evaluation (in MRR) across different feature inputs

Dataset	Layer 1							Layer 2						
	$n_i^h+n_i^t$	$b_i^h+b_i^t$	$d_i^h+d_i^t$	n_i+b_i	n_i+d_i	b_i+d_i	ψ	$n_i^h+n_i^t$	$b_i^h+b_i^t$	$d_i^h+d_i^t$	n_i+b_i	n_i+d_i	b_i+d_i	ψ
FB15k-237	0.3840	0.3158	0.4612	0.6056	0.7776	0.7675	**0.8262**	0.7302	0.5697	0.7607	0.8103	0.8115	0.8321	**0.8056**
NELL-995	0.4118	0.3903	0.4381	0.4030	0.4316	0.4248	**0.4510**	0.4341	0.4158	0.4363	0.4533	0.4604	0.4528	**0.4840**
Health-KG	0.3805	0.3247	0.3827	0.3805	0.3852	0.3827	**0.3846**	0.3805	0.3247	0.3827	0.3805	0.3849	0.3827	**0.3843**

6.4 Attack 3: Graph Reconstruction Performance

Attack 3 performs graph reconstruction by inferring multiple private triples in KGpriv. For each candidate head h and target relation rtarget ∈ Rtarget, A3 computes the similarity between the target vertex and its k-nearest neighbors (k-NN) in the public graph using the feature set φ3 defined in Equation 9. A voting mechanism among k-NN neighbors determines whether each candidate triple exists in KGpriv.

From Table 5 which shows the performance of Attack 3, we notice that A3 achieves strong reconstruction performance using topology alone. Across all knowledge graphs, the attack successfully recovers private triples, with PR-AUC ranging from 0.481 to 0.585 and strong ranking performance (results can be found in Appendix D). This validates Hypothesis 3: topological similarity enables accurate reconstruction.

Furthermore, we remark that unlike Attacks 1 and 2, Attack 3 requires deeper topological context. Level 1 alone yields poor performance across all graphs, demonstrating that immediate neighborhoods remain insufficient for reconstruction. Performance substantially improves with each additional level. We limit our evaluation to three levels because as neighborhood sizes grow exponentially with each level, feature extraction becomes prohibitively expensive. These findings demonstrate that successful reconstruction exploits multiple neighborhoods levels to capture the topological patterns necessary for accurate graph reconstruction.

As opposed to Attacks 1 and 2, Attack 3 performs the best with Health-KG. This is due to the fact that A3 does not use a global classifier anymore and relies on a voting mechanism throughout the neighbors of a vertex (in different levels). Consequently, in this context, local homogeneity seems to be more helpful for Attack 3. Health-KG exhibits high mean of out-degree with low variance (see Figure 2) and hence stabilizes the precision of the voting system. On the other hand, the heterogeneous FB15k-237 with many vertices having few neighbors negatively impacts the precision of the voting.

Table 5: Graph reconstruction (Attack 3) Evaluation (in PR-AUC) across different feature inputs

Dataset	Layer 1					Layer 2					Layer 3				
	$n_i^h n_i^t$	$n_i^h n_i^t b_i$	$n_i^h n_i^t d_i$	$b_i^h b_i^t$	ϕ_3	$n_i^h n_i^t$	$n_i^h n_i^t b_i$	$n_i^h n_i^t d_i$	$b_i^h b_i^t$	ϕ_3	$n_i^h n_i^t$	$n_i^h n_i^t b_i$	$n_i^h n_i^t d_i$	$b_i^h b_i^t$	ϕ_3
FB15k-237	0.229	0.253	0.229	0.238	**0.253**	0.306	0.348	0.342	0.329	**0.377**	0.358	0.381	0.384	0.359	**0.518**
NELL-995	0.095	0.124	0.094	0.118	**0.124**	0.380	0.453	0.454	0.311	**0.457**	0.471	0.478	0.462	0.416	**0.481**
Health-KG	0.640	0.684	0.631	0.362	**0.476**	0.635	0.671	0.513	0.412	**0.536**	0.645	0.674	0.531	0.450	**0.585**

6.5 Summary

This experimental study shows that for Attacks 1 and 2, graph heterogeneity increases their success rate whereas Attack 3 exhibits an opposite behavior and performs better when the graph is more homogeneous. The main reason for this difference originates from the actual design choices of the attacks. Indeed, while A1 and A2 train and use classifiers to learn the correlation between public and

private graphs, A3, due to computational constraints only obtains information about the immediate neighborhood of a target vertex.

Main Takeaway. Attacks exploiting topological features across the entire public graph (i.e., Attacks 1 and 2), are more successful with the increase in heterogeneity in the KG. On the other hand, when the attack can only make use of topological features on a locally bounded neighborhood, its success raises with the KG's homogeneity.

7 Defenses

Having demonstrated that topology alone can accurately disclose private information in KGs, we now investigate potential defense mechanisms. We first study the effectiveness of two state of the art defense strategies against the three newly designed attacks, namely: randomized response, which applies edge-level privacy through probabilistic perturbation, and degree-based k-anonymity, which targets degree distributions. We then design a new solution named Chameleon which specifically perturbs the correlation between public and private graph topology and therefore shows more promising performance. To evaluate the performance of Chameleon and compare it with the two other defense strategies, we implement them on the three KGs described in section 6.1 and measure the effectiveness of Attacks 1 and 3.

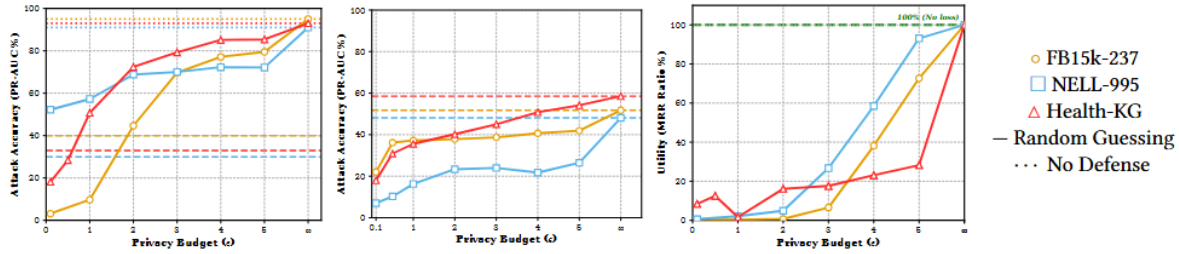
7.1 Randomized Response

Randomized response [23, 25, 32] enables link-level privacy through probabilistic perturbation. To satisfy the paradigm of ε-differential privacy [7], each existing edge is removed and replaced with another new edge in the KG with probability pflip = 1/(eε + 1). This guarantees ε-edge differential privacy: for any two graphs differing by a single edge, the probability ratio of generating the same output is bounded by eε, where smaller ε values provide stronger privacy.

We evaluate randomized response across privacy budgets from ε = 0.1 to ε = 5 where lower values provide stronger privacy guarantees. In figure 4a, we observe that starting from ε = 1, the defense becomes almost ineffective (with PR-AUC>60%) against Attack 1 for NELL and Health-KG. On the other hand, the protection seems to be more effective in the context of FB15k-237. We believe that this is due to distributional characteristics: FB15k-237's broader feature distributions (degree range: 1 to 25+, relation diversity: 2 to 20) are disrupted more effectively by random perturbations. As established in Takeaway 1, high variance makes graphs more vulnerable. This same property makes Randomized response more effective on FB15k-237: random perturbations spread vertices across the wide range, hiding the patterns that attacks use. In contrast, NELL-995's concentrated distributions (median degree: 1, relation diversity: 1) are harder to disrupt. Because most vertices already share similar low values, perturbations cannot separate them enough to hide their relative positions. Attack 2 follows Attack 1's pattern as triple inference depends fundamentally on link inference.

Attack 3 exhibits similar patterns (Figure 4b), with protection degrading at ε ≥ 1 for NELL-995 and Health-KG while FB15k-237 maintains stronger resistance. At strong privacy levels (ε = 0.1), the mechanism provides high privacy guarantees, but at the cost of severely distorting the graph, resulting in an almost random topology with little remaining topological information. This is because the defense

strategy perturbs edges uniformly without targeting the specific correlations between features φ1, φ3 and target relation that enable inference.



(a) Attack 1 (b) Attack 3 (c) Utility

7.2 K-anonymity

Degree-based k-anonymity [8, 12, 20, 38] ensures that every node has the same degree with at least k − 1 other vertices. More formally, a protected graph ensures that:

$$\forall v \in E, \exists S \subseteq E : |S| \geq k \text{ and } deg(u) = deg(v), \forall u \in S.$$

Hence, unlike randomized response, k-anonymity does take topology into account by preventing re-identification through degree indistinguishability.

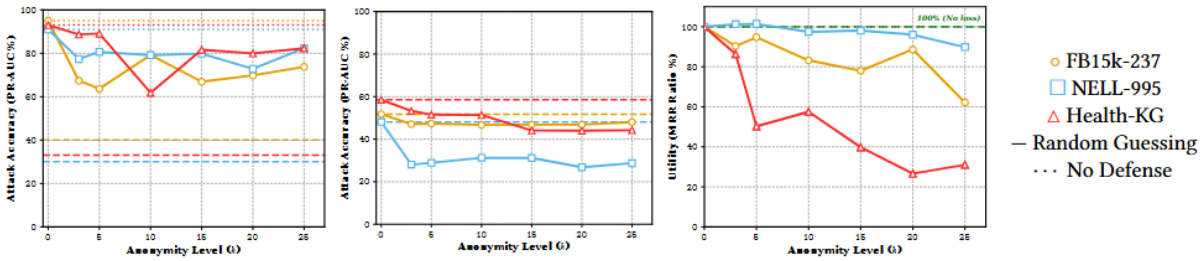
To study the effectiveness of this strategy against Attacks 1 to 3, we have implemented bucket-median anonymization [12] with anonymity levels ranging from k = 5 to k = 25: vertices are sorted by degree and partitioned into buckets of size k, then adjusted to the median degree through edge additions or removals. Vertices below the median receive random edges to non-neighbors with relation types sampled from the existing public graph distribution, while vertices above the median have edges randomly removed. This mechanism equalizes total degree distributions and modifies, implicitly, relation diversity through random edge additions, but does not identify or target which specific topological patterns within φ1 are discriminative for inference.

Figure 3a reveals k-anonymity's limited effectiveness against Attack 1. Rather than declining with increased anonymity, attack accuracy remains persistently high across anonymity levels. FB15k-237 fluctuates between 65% and 73% with no clear downward trend, NELL-995 oscillates between 78% and 85%, and Health-KG shows erratic behavior ranging from 63% to 92%. Even at maximum anonymity with k = 25, all datasets maintain attack success rates between 73% and 83%, far exceeding the random-guessing baselines. Attack 2 inherits Attack 1's limited protection because triple inference depends on link identification.

Figure 3b demonstrates similar limited effectiveness for Attack 3. FB15k-237 and NELL-995 maintain PR-AUC of 47 to 53% across all anonymity levels, showing limited protection. Health-KG drops from 82% to 45%, providing moderate protection.

We conclude that k-anonymity fundamentally fails to provide meaningful protection in this setting. Its uniform anonymization strategy does not target the correlations that enable inference, leaving the core predictive signals largely unaffected. As a result, the topological patterns linking public graph to target

relation remain exploitable. Consequently, even at the highest anonymity level (k = 25), the defense is ineffective, with attacks maintaining high success rates.



(a) Attack 1 (b) Attack 3 (c) Utility

7.3 Chameleon

We propose a new defense strategy named Chameleon that directly targets the correlation between public graph topology and the private graph. Unlike k-anonymity that modifies features uniformly across the public graph, Chameleon first scores vertices based on their topological patterns that correlate with target relations, then applies targeted perturbations to disrupt these correlations. The mechanism operates through three coordinated components that address different attack surfaces.

Given a global privacy budget $B \in [0, 1]$ (representing the fraction of edges to perturb), we allocate budget components where $B = \alpha + \beta + \gamma$: α for lightweight proxy disruption, β for synthetic edge insertion, and γ for relation obfuscation. The lightweight proxy disruption component scores edges by common neighbor count and removes the top $\lfloor \alpha \cdot |E| \rfloor$ edges, degrading connectivity coefficients bi and di that distinguish participants from non-participants. By removing high-scoring edges rather than random edges, this component significantly impacts the correlation that enables inference.

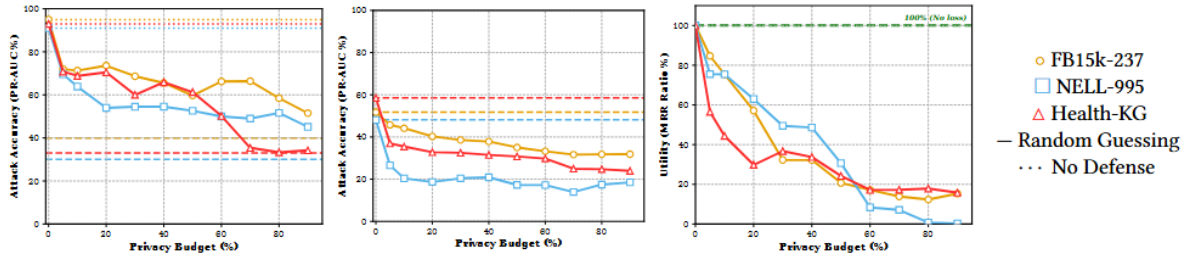
The synthetic edge insertion component targets node pairs that are not directly connected but share common neighbors (distance-2 pairs: vertices separated by exactly one intermediate node). The component scores these pairs by their number of common neighbors, then adds $\lfloor \beta \cdot |E| \rfloor$ synthetic edges with relation labels between pairs having 1 to 3 common neighbors. Unlike k-anonymity's random edge additions, these edges are strategically placed where they appear topologically plausible based on existing connectivity patterns. This inflates degree distributions and connectivity coefficients specifically for vertices whose topological profiles correlate with target relations, thereby disrupting the discriminative signals that Attacks 1 to 3 exploit.

Finally, the relation profile obfuscation component first scores vertices by relation-type distribution frequency, prioritizing less frequent patterns for modification, and then swaps labels on edges incident between each other to the top γ fraction, flattening relation diversity features ρi for vertices whose relation profiles correlate most strongly with private participation.

Our experiments show that Chameleon achieves effective privacy protection across diverse topologies. Figure 5a demonstrates consistent attack effectiveness reduction. At low budget (20%), Attack 1 (and consequently Attack 2) accuracy drops to 54-68%, providing meaningful privacy. At medium budget (50%), attacks reach 48-56%, approaching random guessing. At high budget (80%), attack accuracy

achieves 33-50%, with Health-KG reaching the random guessing baseline. Figure 5b shows comparable effectiveness for Attack 3: PR-AUC drops to 35-69% (20%), 20-39% (50%), and 19-32% (80%).

Unlike topology-agnostic defenses that do not fully address the public-private correlation (k-anonymity providing limited protection despite targeting degree), Chameleon achieves protection by directly disrupting the topological patterns that enable inference. This topology-aware design targets the precise correlation between the public graph and private one, degrading attack effectiveness proportionally to the privacy budget regardless of underlying graph topology. In Appendix G we formally describe the Chameleon algorithm.



(a) Attack 1 (b) Attack 3 (c) Utility

7.4 Discussion: Privacy vs Utility

The fundamental challenge in data privacy lies in balancing sensitive information protection with graph utility preservation. To evaluate utility, we propose to study link prediction tasks across defended graphs. Specifically, we train a link prediction model on each one of them and measure its Mean Reciprocal Rank (MRR) on a held-out test set. Utility is then quantified as the MRR ratio = MRRdefended / MRRoriginal, where values closer to 1 indicate better utility preservation and values closer to 0 indicate severe utility degradation. This metric captures how well the defended graph maintains its predictive structure for downstream tasks.

Randomized response exhibits variable effectiveness across topologies. At $\varepsilon = 0.1$, FB15k-237 and Health-KG achieve strong privacy but utility drops to 7-11%. NELL-995 shows different behavior: utility degrades to 8% yet attacks retain 53% effectiveness. This occurs because NELL-995's distribution shows extreme concentration with most vertices at degree 1 and minimal variance (Figure 6a), allowing random perturbation to preserve relative node rankings. In contrast, FB15k-237's heavy-tailed distribution with wide spread (mean: 18.4, range: 1-1400+) is disrupted more effectively as noise shifts vertices across broader value ranges. At $\varepsilon \geq 5$, utility recovers but attacks return to baseline.

K-anonymity provides ineffective trade-offs across all distribution types. NELL-995, with its narrow, concentrated distributions (Figure 6), maintains 90-100% utility with minimal protection (78-85% attacks) because uniform degree equalization within already concentrated ranges fails to disrupt discriminative ordering. Health-KG loses utility (31%) without privacy gain (63-92% attacks) despite moderate distribution variance. This occurs because uniform modifications do not target the specific topology correlations that enable inference, regardless of underlying distribution shape.

Chameleon demonstrates fundamentally different characteristics via topology-aware scoring. Rather than uniform perturbations, it identifies high-risk topological patterns and disrupts them selectively. As budget

increases, attack effectiveness declines consistently while utility degrades proportionally. At low privacy budget (B = 20%), meaningful privacy emerges (attacks: 54-68%) with acceptable utility cost (45-65%). At medium budget (50%), near-balanced trade-offs appear (attacks: 48-56%, utility: 8-35%). At high budget (80%), strong privacy protection arrives (attacks: 33-50%) with modest utility retention (3-17%). Protection scales consistently across diverse topologies.

Table 6 quantifies privacy-utility trade-offs at three privacy levels defined by Attack 1 performance: low privacy (80-90%), medium privacy (60%), and high privacy (5-10%, above random guessing). At low privacy levels, all defenses maintain high utility (Chameleon: 58-84%, RR: 22-70%, k-anonymity: 50-96%). At medium privacy levels, Chameleon achieves balanced trade-offs (36-50% utility retention) while RR drops substantially (7-18%). K-anonymity shows no achievable configuration for most graphs at this level. At high privacy levels, Chameleon retains modest but usable utility (10-19%), whereas RR's utility collapses to near-zero (0.2-5%). These results demonstrate that Chameleon's topology-aware targeting enables superior utility retention at equivalent privacy levels compared to baseline defenses. The key insight is that effective privacy requires selectively disrupting the specific topological correlations that enable inference, rather than applying uniform modifications that either fail to disrupt discriminative patterns or destroy utility entirely.

Table 6: Privacy-utility trade-off

Dataset	Defense	Utility (Low priv.)	Utility (Medium priv.)	Utility (High priv.)
FB15k-237	CHAMELEON	92%	68%	15%
	RR	70%	7%	0.2%
	K-anon*	90%	Not achievable at any k	
NELL-995	CHAMELEON	95%	20%	5%
	RR	90%	5%	2%
	K-anon*	96%	Not achievable at any k	
Health-KG	CHAMELEON	93%	38%	18%
	RR	18%	10%	1.69%
	K-anon*	50%	Not achievable at any k	

** Attack 1 performance remains above 60% (PR-AUC) at any K.*

8 Conclusion

In this work, we have conducted a novel analysis on vulnerabilities that may arise from the use of knowledge graphs with privacy-sensitive data. Our study has shown that the underlying topology of the public part of the KG retains significant information about edges in the private part and demonstrated such information alone is sufficient to result in data leakage through our three incremental attacks. We have also designed a novel privacy-preserving mechanism called Chameleon that mitigates these vulnerabilities by actually perturbing the public graph while taking into account the most impactful topological features.

This work opens the door to several interesting new avenues for research: Firstly, in our study, the KG is considered as being static. It would be interesting to evaluate our attacks in the context of spatio-temporal KG where vertices, edges, and relations can be modified with time. Furthermore, the problem where KGs are distributed among several parties can result in potentially new attack surfaces. On the defense side,

similarly, it would be interesting to study the suitability of Chameleon to the dynamic and distributed settings.

Ethical Considerations

We believe our work does not incur into any ethical concerns.

Open Science

Our work adheres to the Open Science framework to ensure transparency and reproducibility. All knowledge graphs, implementation code, and experimental configurations are publicly available in: https://anonymous.4open.science/r/Exposed_by_Design-402B/.

AI Use

We used Grammarly for grammar verification and an LLM (specifically Claude, Anthropic) to assist with writing README files and code comments for our GitHub repository. No AI tools were used for brainstorming, literature review, references, experimental design, implementation, or analysis. All intellectual contributions are by the human authors.

References

[1] Sofiane Azogagh, Zelma Aubin Birba, Josée Desharnais, Sébastien Gambs, Marc-Olivier Killijian, and Nadia Tawbi. 2025. GRAND: Graph Reconstruction from Potential Partial Adjacency and Neighborhood Data. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (Toronto ON, Canada) (KDD '25). Association for Computing Machinery, New York, NY, USA, 47–58. doi:10.1145/3711896.3736988

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (Lake Tahoe, Nevada) (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, 2787–2795.

[3] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam Hruschka, and Tom Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. Proceedings of the AAAI Conference on Artificial Intelligence 24, 1 (Jul. 2010), 1306–1313. doi:10.1609/aaai.v24i1.7519

[4] Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning. ACM, 233–240. doi:10.1145/1143844.1143874

[5] Phuc Do and Truong H. V. Phan. 2022. Developing a BERT based triple classification model using knowledge graph embedding for question answering system. Applied Intelligence (2022), 636–651. doi:10.1007/s10489-021-02460-w

[6] Vasisht Duddu, Antoine Boutet, and Virat Shejwalkar. 2021. Quantifying Privacy Leakage in Graph Embedding. In MobiQuitous 2020 (Darmstadt, Germany). ACM, New York, NY, USA, 76–85. doi:10.1145/3448891.3448939

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In Theory of Cryptography, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 265–284. doi:10.1007/11681878_14

[8] Alessandro Epasto, Hossein Esfandiari, Vahab Mirrokni, and Andres Munoz Medina. 2024. Smooth Anonymity for Sparse Graphs. In Companion Proceedings of the ACM Web Conference 2024. ACM, 621–624. doi:10.1145/3589335.3651561

[9] Jie Fu, Yuan Hong, Zhili Chen, and Wendy Hui Wang. 2025. Safeguarding Graph Neural Networks against Topology Inference Attacks. 15 pages. doi:10.1145/3719027.3765173

[10] Neil Zhenqiang Gong and Bin Liu. 2016. You Are Who You Know and How You Behave: Attribute Inference Attacks via Users' Social Friends and Behaviors. In 25th USENIX Security Symposium. USENIX Association, Austin, TX, 979–995.

[11] Neil Zhenqiang Gong and Bin Liu. 2018. Attribute Inference Attacks in Online Social Networks. ACM Trans. Priv. Secur. 21, 1, Article 3 (Jan. 2018), 30 pages. doi:10.1145/3154793

[12] A. Mohammed Hanafy, Sherif Barakat, and Amira Rezk. 2021. An Improved K-anonymization Approach for Preserving Graph Structural Properties. International Journal of Advanced Computer Science and Applications 12, 9 (2021). doi:10.14569/IJACSA.2021.0120924

[13] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2021. Stealing Links from Graph Neural Networks. In 30th USENIX Security Symposium. USENIX Association, 2669–2686.

[14] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. 2021. Node-Level Membership Inference Attacks Against Graph Neural Networks. arXiv:2102.05429 [cs.CR]

[15] Anh-Tu Hoang, Barbara Carminati, and Elena Ferrari. 2024. Protecting Privacy in Knowledge Graphs With Personalized Anonymization. IEEE Transactions on Dependable and Secure Computing 21, 4 (2024), 2181–2193. doi:10.1109/TDSC.2023.3300360

[16] Yuke Hu, Wei Liang, Ruofan Wu, Kai Xiao, Weiqiang Wang, Xiaochen Li, Jinfei Liu, and Zhan Qin. 2023. Quantifying and Defending against Privacy Threats on Federated Knowledge Graph Embedding. In Proceedings of the ACM Web Conference 2023 (Austin, TX, USA). ACM, New York, NY, USA, 2306–2317. doi:10.1145/3543507.3583450

[17] Yuke Hu, Yang Wang, Jian Lou, Wei Liang, Ruofan Wu, Weiqiang Wang, Xiaochen Li, Jinfei Liu, and Zhan Qin. 2025. Privacy Risks of Federated Knowledge Graph Embedding: New Membership Inference Attacks and Personalized Differential Privacy Defense. IEEE Transactions on Dependable and Secure Computing 22, 3 (2025), 2788–2805. doi:10.1109/TDSC.2024.3522025

[18] Bhushan Kotnis and Vivi Nastase. 2017. Analysis of the Impact of Negative Sampling on Link Prediction in Knowledge Graphs. doi:10.48550/arXiv.1708.06816

[19] Thanh Le, Nam Le, and Bac Le. 2023. Knowledge graph embedding by relational rotation and complex convolution for link prediction. Expert Syst. Appl. 214, C, 23 pages. doi:10.1016/j.eswa.2022.119122

[20] Kun Liu and Evimaria Terzi. 2008. Towards identity anonymization on graphs (SIGMOD '08). ACM, New York, NY, USA, 93–106. doi:10.1145/1376616.1376629

[21] Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip Yu. 2021. KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning. Proceedings of the AAAI Conference on Artificial Intelligence 35, 6418–6425. doi:10.1609/aaai.v35i7.16796

[22] Menglin Lu, Yujie Zhang, Suixia Zhang, Hanrui Shi, and Zhengxing Huang. 2023. Knowledge-aware patient representation learning for multiple disease subtypes. Journal of Biomedical Informatics 138 (2023), 104292. doi:10.1016/j.jbi.2023.104292

[23] Zhan Qin, Ting Yu, Yin Yang, Issa Khalil, Xiaokui Xiao, and Kui Ren. 2017. Generating Synthetic Decentralized Social Graphs with Local Differential Privacy (CCS '17). ACM, New York, NY, USA, 425–438. doi:10.1145/3133956.3134086

[24] Roan Spadazzi, Vladyslav Husak, and Andrea Policano. 2025. The Human Microbiome and Person-to-Person Interactions Knowledge Graph. KnowDive, University of Trento.

[25] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally Differentially Private Protocols for Frequency Estimation. In 26th USENIX Security Symposium. USENIX Association, Vancouver, BC, 729–745.

[26] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Anchorage, AK, USA). ACM, New York, NY, USA, 950–958. doi:10.1145/3292500.3330989

[27] Xiuling Wang and Wendy Hui Wang. 2022. Group Property Inference Attacks Against Graph Neural Networks. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (Los Angeles, CA, USA). ACM, New York, NY, USA, 2871–2884. doi:10.1145/3548606.3560662

[28] Yu Wang, Lifu Huang, Philip S. Yu, and Lichao Sun. 2022. Membership Inference Attacks on Knowledge Graphs. arXiv:2104.08273 [cs.AI]

[29] Yixin Wu, Xinlei He, Pascal Berrang, Mathias Humbert, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2024. Link Stealing Attacks Against Inductive Graph Neural Networks. Proceedings on Privacy Enhancing Technologies 2024, 4 (2024), 818–839. doi:10.56553/popets-2024-0143

[30] Xiayu Xiang, Zhongru Wang, Yan Jia, and Binxing Fang. 2019. Knowledge Graph-Based Clinical Decision Support System Reasoning: A Survey. In 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC). 373–380. doi:10.1109/DSC.2019.00063

[31] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Product Knowledge Graph Embedding for E-commerce. In Proceedings of the 13th International Conference on Web Search and Data Mining (Houston, TX, USA). ACM, New York, NY, USA, 672–680. doi:10.1145/3336191.3371778

[32] Qingqing Ye, Haibo Hu, Man Ho Au, Xiaofeng Meng, and Xiaokui Xiao. 2022. LF-GDPR: A Framework for Estimating Graph Metrics With Local Differential Privacy. IEEE Transactions on Knowledge and Data Engineering 34, 10 (2022), 4905–4920. doi:10.1109/TKDE.2020.3047124

[33] Hanyang Yuan, Jiarong Xu, Cong Wang, Ziqi Yang, Chunping Wang, Keting Yin, and Yang Yang. 2024. Unveiling Privacy Vulnerabilities: Investigating the Role of Structure in Graph Data. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (Barcelona, Spain). ACM, New York, NY, USA, 4059–4070. doi:10.1145/3637528.3672013

[34] Oualid Zari, Chuan Xu, Javier Parra-Arnau, Ayse Unsal, and Melek Önen. 2024. Link Inference Attacks in Vertical Federated Graph Learning. In 2024 Annual Computer Security Applications Conference (ACSAC). IEEE Computer Society, Los Alamitos, CA, USA, 761–777. doi:10.1109/ACSAC63791.2024.00068

[35] He Zhang, Bang Wu, Shuo Wang, Xiangwen Yang, Minhui Xue, Shirui Pan, and Xingliang Yuan. 2023. Demystifying Uneven Vulnerability of Link Stealing Attacks against Graph Neural Networks. In Proceedings of the 40th International Conference on Machine Learning (PMLR, Vol. 202). 41737–41752.

[36] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, and Enhong Chen. 2021. GraphMI: Extracting Private Graph Data from Graph Neural Networks. In Proceedings of IJCAI-21. 3749–3755. doi:10.24963/ijcai.2021/516

[37] Elena Zheleva and Lise Getoor. 2007. Preserving the privacy of sensitive relationships in graph data. In Proceedings of the 1st ACM SIGKDD International Conference on Privacy, Security, and Trust in KDD (San Jose, CA, USA). Springer-Verlag, Berlin, Heidelberg, 153–171. doi:10.1007/978-3-540-78478-4_9

[38] Bin Zhou, Jian Pei, and WoShun Luk. 2008. A brief survey on anonymization techniques for privacy preserving publishing of social network data. SIGKDD Explor. Newsl. 10, 2 (Dec. 2008), 12–22. doi:10.1145/1540276.1540279

[39] Enyuan Zhou, Song Guo, Zhixiu Ma, Zicong Hong, Tao Guo, and Peiran Dong. 2024. Poisoning Attack on Federated Knowledge Graph Embedding. In Proceedings of the ACM Web Conference 2024 (Singapore, Singapore). ACM, New York, NY, USA, 1998–2008. doi:10.1145/3589334.3645422