De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs

Jianwei Qian
Illinois Tech
Chunhong Zhang
BUPT

Xiang-Yang Li
USTC, Illinois Tech
Linlin Chen
Illinois Tech
Outline

- Background
- Prior Work
- Our Work
- Conclusion

De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs
Background

- Tons of social network data
- Released to third-parties for research and business
- Though user IDs removed, attackers with prior knowledge can de-anonymize them. → privacy leak
Attacking Process

Sanitization

ID Removal

Prior k.g.

De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs
De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs
Attack Stage 1
De-Anonymization

Which is Alice?
Which is Bob?

Direct privacy leak

De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs
Attack Stage 2
Privacy Inference

- **Correlations** between attributes/users
  - Higher education => higher salary
  - Colleagues => same company
  - Common hobbies => friends

- **Infer new info** that is not published

**Indirect privacy leak**
What Do We Want to Do?

To understand

How privacy is leaked to the attacker
Outline

Background

Prior Work

Our Work

Conclusion

De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs
De-anonymize one user

Never ending!

Assume specific prior knowledge!

- Degree attack [SIGMOD'08]
- 1-neighborhood attack [INFOCOM’13]
- 1*-neighborhood attack
- Community re-identification [SDM’11]
- k-structural diversity
Prior Work

De-anonymize **all the users**
- Graph mapping based de-anonymization
  [WWW’07, S&P’09, CCS’12, COSN’13, CCS’14, NDSS’15]

**Attacker holds an auxiliary SN that overlaps with the published SN**

Twitter

Flickr
Limitations

• Assume attacker has specific prior knowledge
  – We assume diverse and probabilistic knowledge

• Focus on de-anonymization only. How attacker infers privacy afterwards is barely discussed
  – We consider it as 2nd attacking step!
Outline

Background

Prior Work

Our Work

Conclusion
Goals

• To construct a comprehensive and realistic model of the attacker’s knowledge

• To use this model to depict how privacy is leaked.
• Hard to build such an expressive model, given that the attacker has *various prior knowledge*

• Hard to simulate attacking process, since the attacker has *various techniques*
Solution

Use **knowledge graph** to model attacker’s knowledge
Knowledge Graph

- Knowledge => directed edge
- Each edge has a confidence score

De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs
What’s Privacy?

• Every edge is privacy
• Privacy is leaked when $|c_p(e) - c_q(e)| > \theta(e)$

De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs
De-Anonymization

Prior knowledge $G_p$

Anonymized graph $G_a$

$\text{argmax} \ Sim_\pi(G_p, G_a)$

$Sim_\pi(G_p, G_a) = \sum_{(i,j) \in \pi} S(i,j)$,

$S$ is node similarity function
Node Similarity

- **Attribute Similarity**
  - Use Jaccard index to compare attribute sets

- **Relation similarity**
  - Inbound neighborhood
  - Outbound neighborhood
  - l-hop neighborhood

\[
S_R(i, j) = w_i S_i(i, j) + w_o S_o(i, j) + w_l S_l(i, j)
\]

\[
S(i, j) = w_A S_A(i, j) + (1 - w_A) S_R(i, j)
\]
Problem Transformation

Mapping => Max weighted bipartite matching

Naïve method:

Huge complexity!

$G_p$  $\longrightarrow$  $G_a$

$n_p$  $\longrightarrow$  $n_a$ (millions)
Top-$k$ Strategy

Suppose $k=2$

Alice

$G_p$

$G_a$

$n_p$

$n_a (\text{millions})$
How to Choose Top-k Candidates?

- **Intuition**
  - If two nodes match, their neighbors are also very likely to match.

- **Perform BFS on** $G_p$
### Complexity Analysis

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building Bipartite</td>
<td>$n_p n_a$</td>
<td>$O \left( (n_p + n_a) n_p^2 n_a \right)$</td>
</tr>
<tr>
<td>Finding Matching</td>
<td>$O \left( (n_p + n_a) n_p^2 n_a \right)$</td>
<td>$O \left( (n_p + n_a)^2 \right)$</td>
</tr>
</tbody>
</table>

- **Naïve method**: $n_p n_a$
- **Top-$k$ strategy**: $\ll n_p n_a$

**Complexity greatly reduced!**
Tradeoff

- $k$ balances accuracy and complexity
- $k = 10$ is enough to achieve high accuracy
Privacy inference

Predict **new edges** in knowledge graph

De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs
Path Ranking Algorithm

• Proposed by Ni Lao et al. in 2011 for a different topic

• Correlations => “rules” => paths
• Logistic regression
Experiments

• Datasets
  – Google+, Pokec

| Dataset | $|\mathcal{V}_U|$ | $|\mathcal{V}_A|$ | $|\mathcal{E}_{UU}|$ | $|\mathcal{E}_{UA}|$ | $|\mathcal{E}_{AA}|$ |
|---------|----------------|----------------|-----------------|----------------|----------------|
| Google+ | 107,614        | 15,691         | 13,673,453      | 378,880        | 2,262          |
| Pokec   | 306,568        | 576            | 2,822,492       | 1,532,840      | 38             |

• Steps
  – Generate $G_a$
  – Generate $G_p$
  – Run the algorithms
De-Anonymization Results

Metrics: accuracy, run time

De-anonymize about 60% of users
Privacy Inference Results

Metrics: hit@k, MRR (*Mean reciprocal rank*)

Infers much more privacy than random guess
Outline

Background

Prior Work

Our Work

Conclusion
Conclusion

We have

• Applied knowledge graphs to model the attacker’s prior knowledge

• Studied the attack process: de-anonymization & privacy inference

• Designed methods to perform attack

• Done simulations and evaluations on two real world social networks
Future work

• Effective construction of the bipartite for large scale social networks

• Impact of adversarial knowledge on de-anonymizability

• Fine-grained privacy inference on the knowledge graph
Thank you!

Jianwei Qian
jqian15@hawk.iit.edu
https://sites.google.com/site/jianweiqianshomepage