ILLINOIS INSTITUTE OF TECHNOLOGY

Privacy Inference on Knowledge Graphs: Hardness and Approximation

Jianwei Qian, Shaojie Tang, Huiqi Liu, Taeho Jung, Xiang-Yang Li

Illinois Tech, UT Dallas, USTC

Outline

- Background & Related Work
- Problem Formulation & Analysis
- Heuristic Algorithm
- Conclusion

Privacy leak is real!

AOL's disturbing glimpse into users' lives

Release of three-month search histories of about 650,000 users provides rare glimpse into their private lives.

THE WALL STREET JOURNAL.

Subscribe | Sign In

Chinese Online Travel Company Ctrip Hacked

U.S.-listed company's service disrupted after attack by 'unidentified sources'

Uber accidentally leaks personal data for hundreds of drivers

by Rich McCormick | Oct 14, 2015, 2:37am EDT

SHARE | TWEET | LINKEDIN

Tip of the iceberg!

Privacy inference

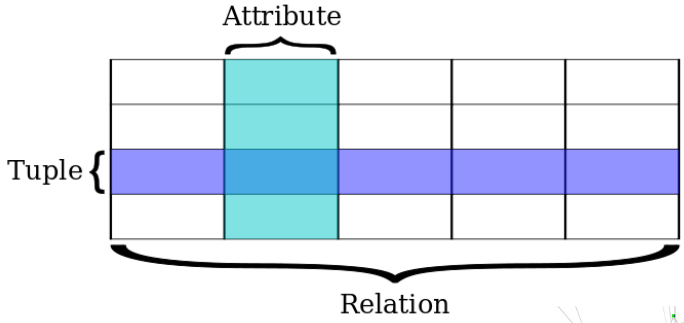

Background knowledge
(Prior knowledge)

The attacker can use background knowledge to infer users' sensitive information. E.g.

- Higher education => higher salary
- Colleagues=> same company
- Common hobbies => friends

Previous privacy attacks

- Relational data
  - Data publishing
  - Statistical query



- Graph data
  - De-anonymization
  - Privacy learning

- Other data forms
  - Spatio-temporal data, genome data, multimedia data

Outline

- Background & Related Work
- Problem Formulation & Analysis
- Heuristic Algorithm
- Conclusion

Our goal

To model the privacy inference attack and reveal its essence from a general view

(differs from most of previous work)

Challenges

- Privacy is difficult to define and privacy leakage is hard to quantify
- It is challenging to apply one single model to various data forms
- Privacy inference is hard to model because there are a large diversity of attack techniques

Overview of the solution

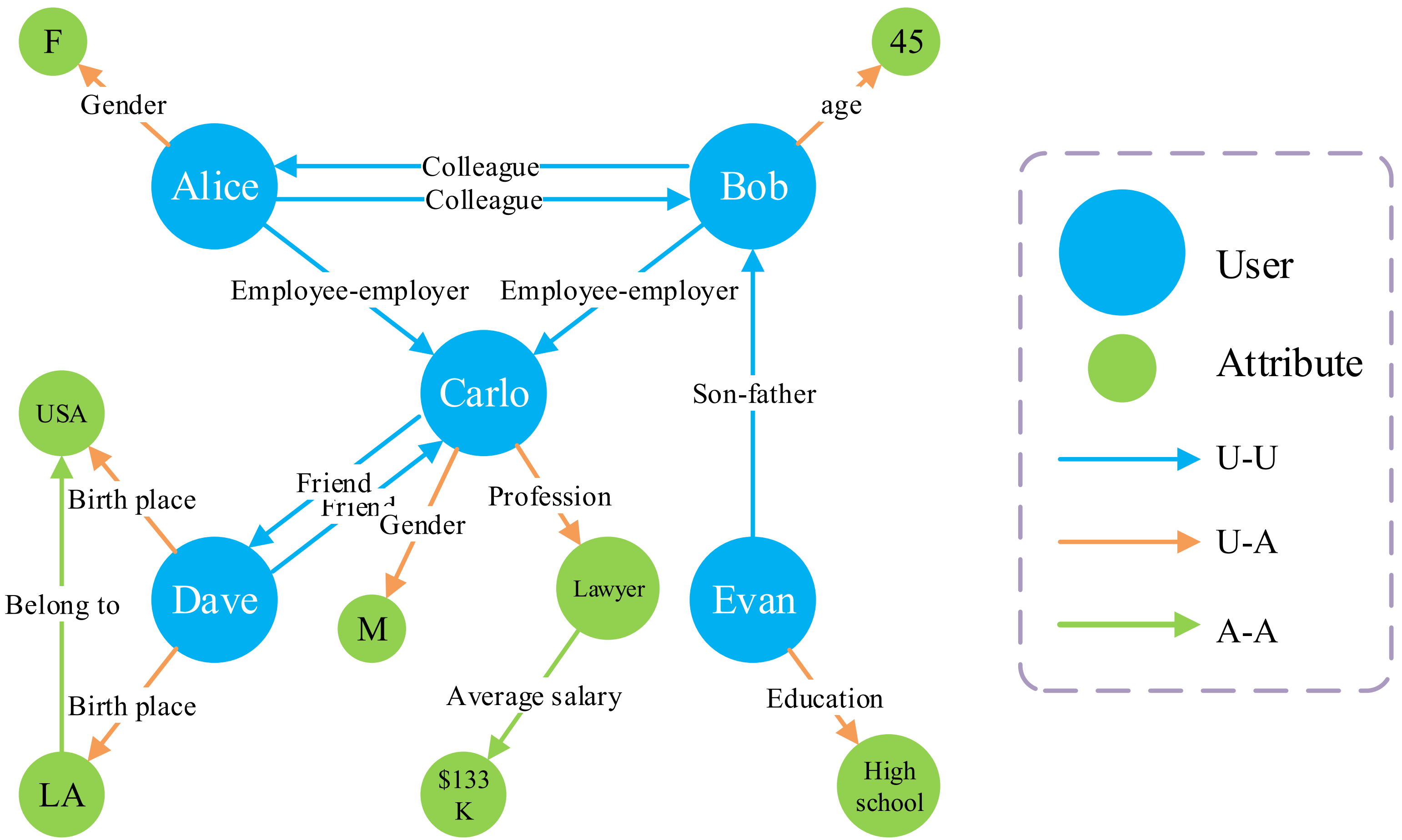
1. Model the attacker's knowledge by a knowledge graph
2. Abstract four base cases of privacy inference
3. Formulate privacy inference and prove the hardness
4. Design approximation algorithm which reflects network evolution

Knowledge graph

A heterogeneous graph of all kinds of entities and their relations related to a specific domain or topic

- E.g. Freebase, Wikidata, Dbpedia, YAGO, NELL, Google's Knowledge Graph, Facebook's Entities Graph

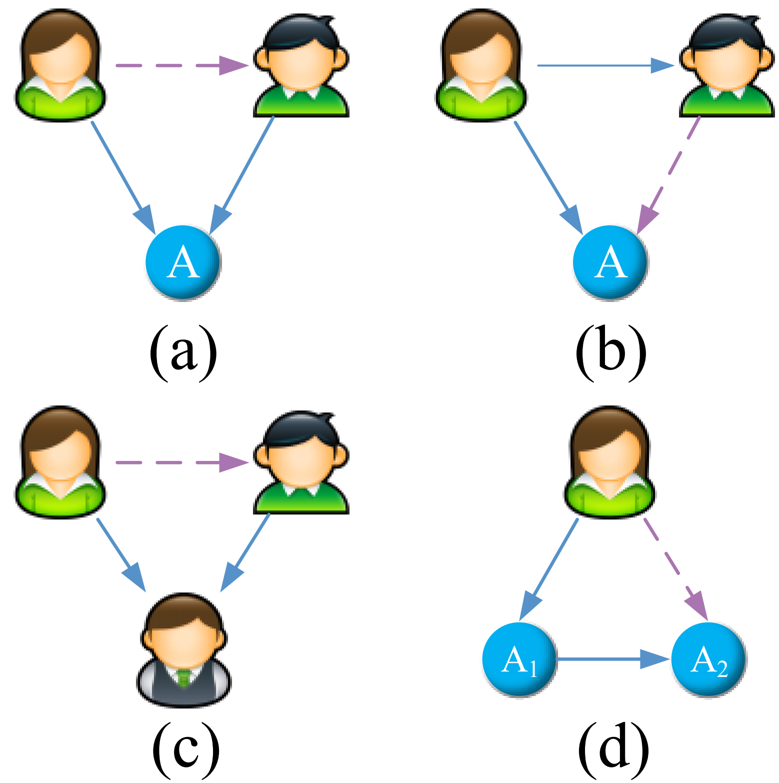
Our knowledge graph



Each edge is associated with a probability, indicating the attacker's confidence

Privacy inference base cases

- Triangle inference
  - A common neighbor

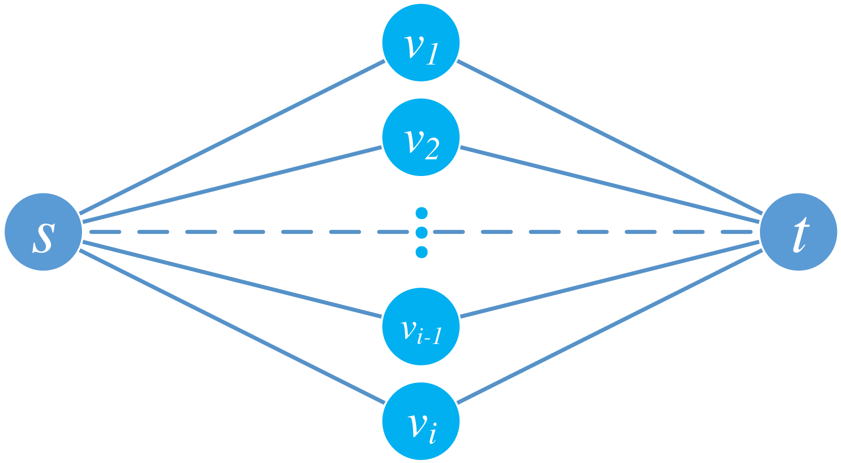


For Fig (a), we have

$$\Pr(e_{u_1u_2}) = \Pr(e_{u_1a})\Pr(e_{u_2a})\Pr(e_{u_1u_2}|e_{u_1a},e_{u_2a})$$

Inference probability

Multiple common neighbors



To infer the relation between s and t

- Multiple base cases are combined altogether if they have multiple common neighbors
- Inference probability $\Pr(e_{st}|N_{st})$ is computed by aggregation

Assumption about triangle inference

- To infer an unknown edge e_{st}:
  - If s, t have common neighbor(s), then e_{st} exists with a probability (the inference probability);
  - If there is no path connecting s, t, then e_{st} must not exists;
  - If s, t do not have a common neighbor but there is a path connecting them, the status of e_{st} is TBD.

Privacy inference

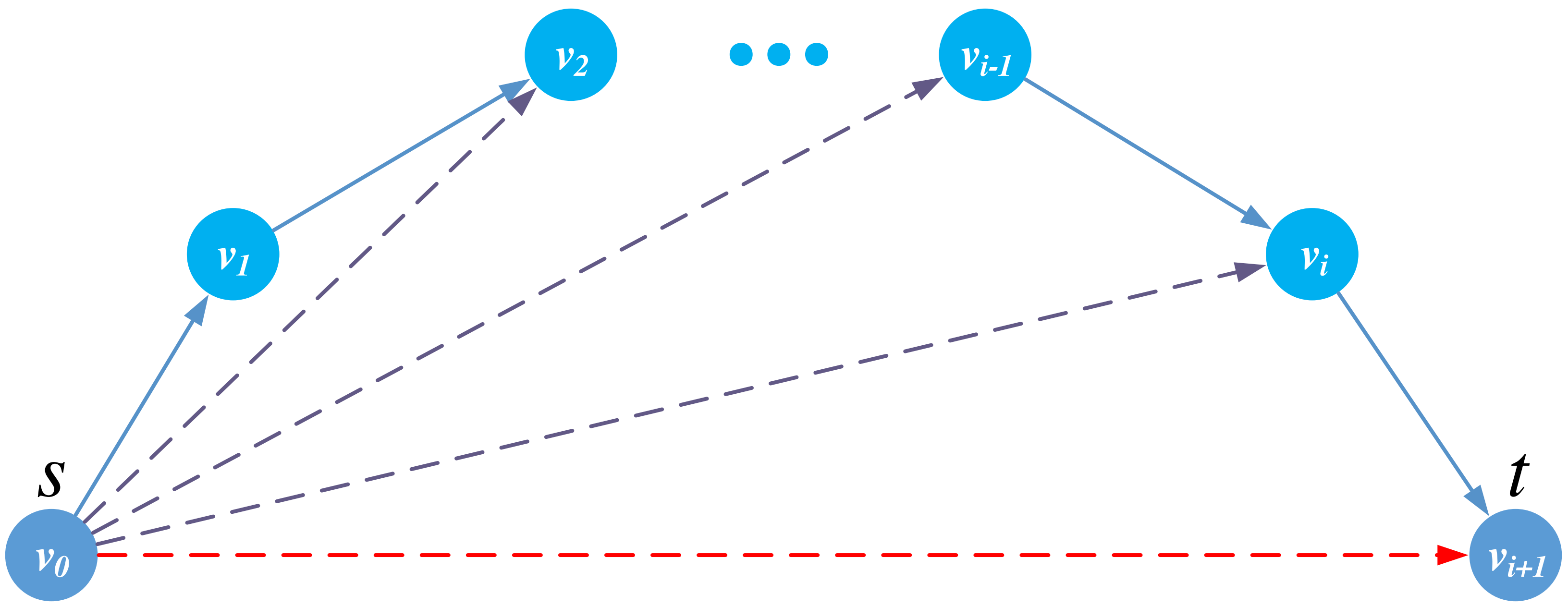
- Is the problem of computing $p(e_{st})$, the probability that an unknown edge e_{st} exists.

Definition 2 (Privacy inference): Given a knowledge graph $G = (U \cup A, E, P)$, privacy inference is the problem of computing $p(e_{s,t})$ for any unknown edge $e_{st} \in E$ where $s \in U$ and $t \in U \cup A$, given the inference probabilities $P(e_{st} \mid N_{st})$, $\forall N_{st} \subseteq U \cup A \backslash \{s, t\}$. We denote this problem as $\mathrm{PI}(G, s, t)$.

#P-hardness

The #P-hard “s-t reliability” problem can be reduced to our problem.

Please see detailed proof in our paper.



Outline

- Background & Related Work
- Problem Formulation & Analysis
- Heuristic Algorithm
- Conclusion

Algorithm

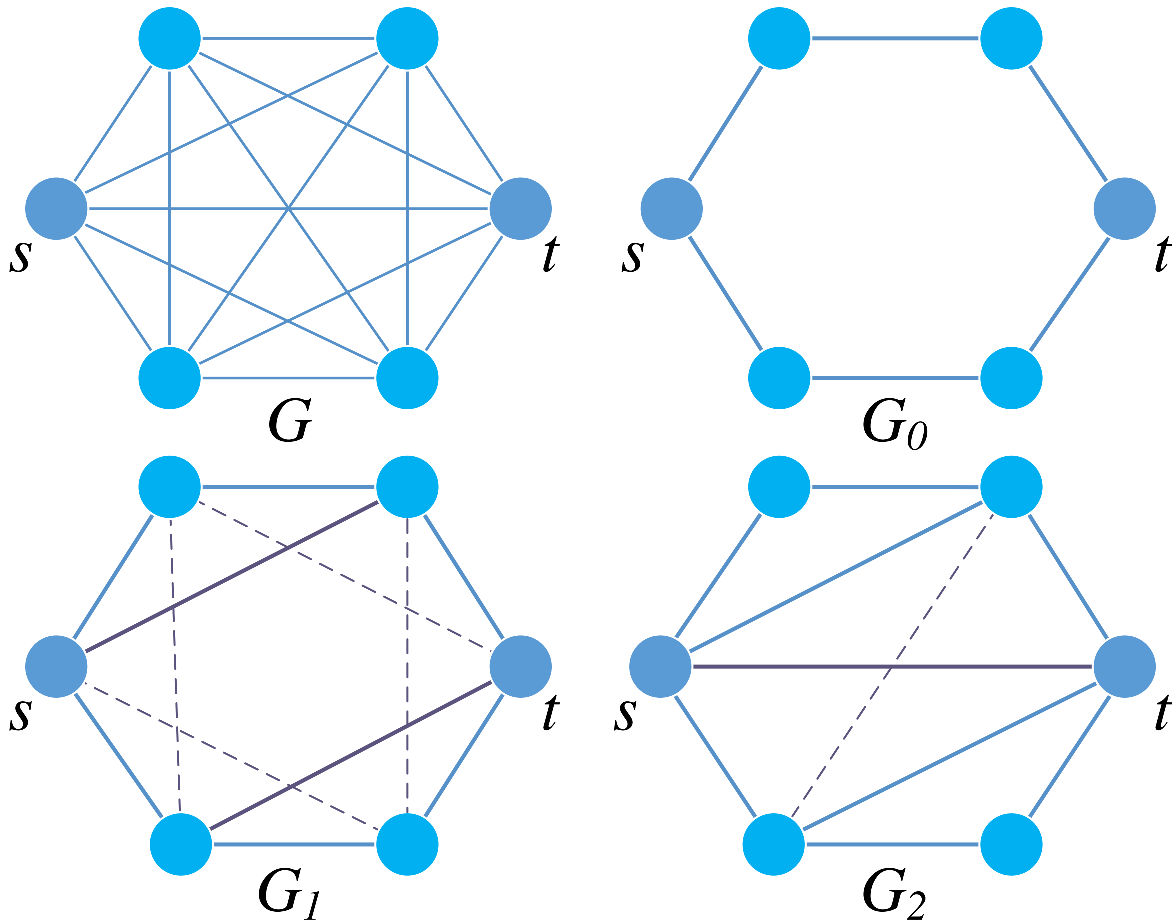
- A iterative algorithm based on Monte Carlo simulation

Generate conjecture graph G_0 from knowledge graph G.

Flip a coin for each candidate pair to decide whether to add an edge.

Stop if e_{st} is inferred or no more edges can be added; redo Step 2 otherwise.

An example



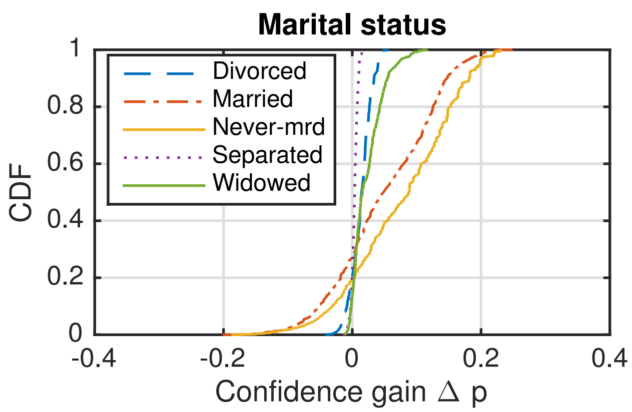
Simulations

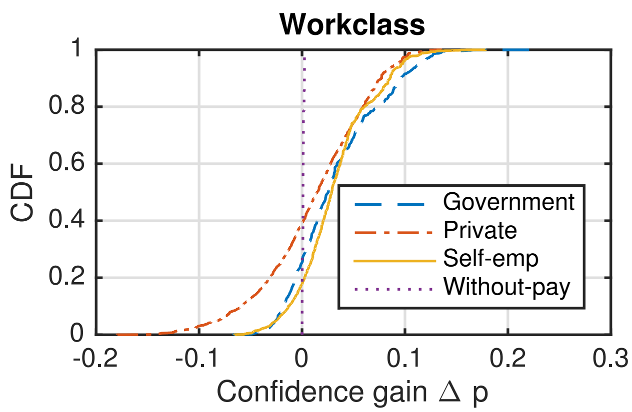
- Datasets
  - Relational: Adult
  - Social network: Pokec
- Knowledge graph construction
  - Edge probabilities are synthetic: uniform, Gaussian
- Inference probabilities are set by statistics

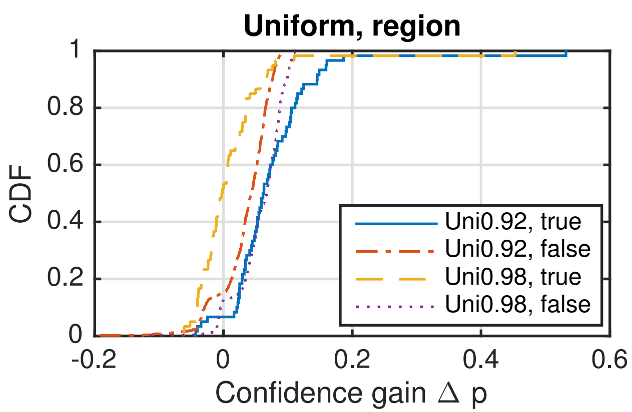
- Metric
  - confidence gain

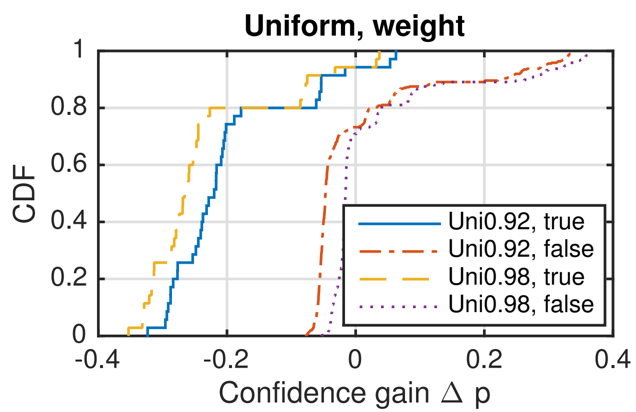
$$\Delta p(e_{st}) = I(e_{st})(p'(e_{st}) - p_0(e_{st}))$$

Results


Marital status
CDF
Confidence gain Δ p
Divorced
Married
Never-mrd
Separated
Widowed
-0.4
-0.2
0
0.2
0.4
0
0.2
0.4
0.6
0.8
1


Workclass
CDF
Confidence gain Δ p
Government
Private
Self-emp
Without-pay
-0.2
-0.1
0
0.1
0.2
0.3
0
0.2
0.4
0.6
0.8
1


Uniform, region
CDF
Confidence gain Δ p
Uni0.92, true
Uni0.92, false
Uni0.98, true
Uni0.98, false
-0.2
0
0.2
0.4
0.6
0
0.2
0.4
0.6
0.8
1


Uniform, weight
CDF
Confidence gain Δ p
Uni0.92, true
Uni0.92, false
Uni0.98, true
Uni0.98, false
-0.4
-0.2
0
0.2
0.4
0
0.2
0.4
0.6
0.8
1

Outline

- Background & Related Work
- Problem Formulation & Analysis
- Heuristic Algorithm
- Conclusion

Summary of contributions

We have

- analyzed the nature of background knowledge and model it on a knowledge graph
- formulated privacy inference and proved its #P-hardness
- designed a heuristic algorithm to approximate it
- done simulations on real world datasets to show the effectiveness of our algorithm

Thank you!