LinkMirage: Enabling Privacy-preserving Analytics on Social Relationships

Changchang Liu, Prateek Mittal
Email: cl12@princeton.edu, pmittal@princeton.edu
Princeton University

February 23, 2016

Social relationships

(a)

(b)

Third party applications rely on users' social relationships:

- E-commerce
- Spam detection
- Anonymous communication
- Sybil defenses

Social relationships are very sensitive!

Social relationships represent

- Trusted friendships
- Important interactions
- Even more, business relations, etc.

How to balance utility and privacy?

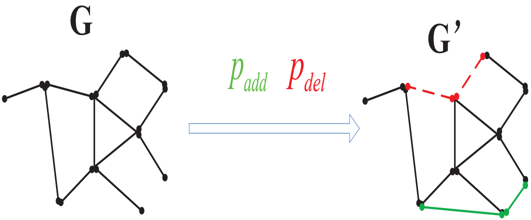
Protect privacy of sensitive social relationships

Preserve utility of obfuscated social relationships for real-world applications

Previous work of link privacy mechanisms

To protect link privacy, previous work

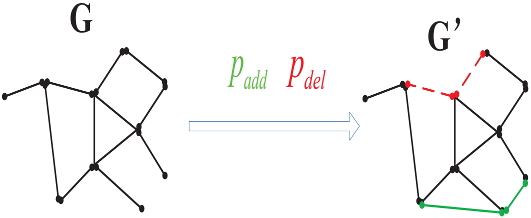
- obfuscate social relationships through link additions/deletions



Limitations of previous link privacy mechanisms

To protect link privacy, previous work

- obfuscate social relationships through link additions/deletions

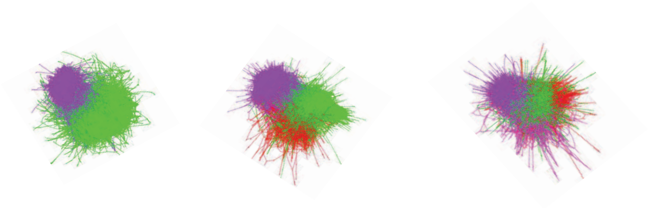


However, previous work

- only consider graph data where the links are static

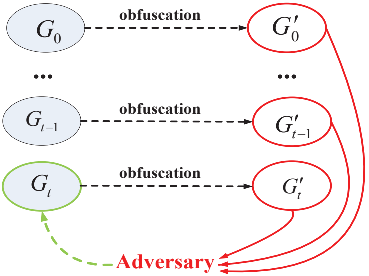
However, social networks are dynamic

Temporal Facebook dataset (every three months) with 46,952 users and 876,993 edges



However, social networks are dynamic

An adversary can combine the previously perturbed graphs together



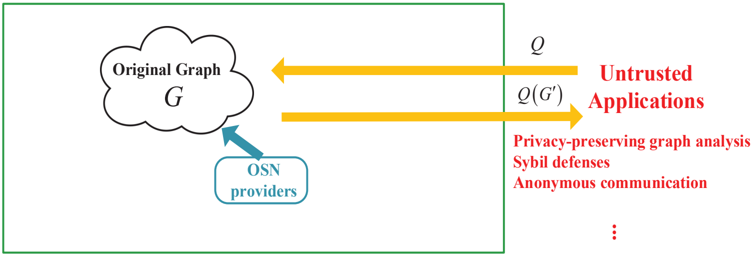
Our Objective

- Balance privacy and utility
- Handle both the static and dynamic social network topologies
- Provide rigorous privacy guarantees
- Useful in real-world applications

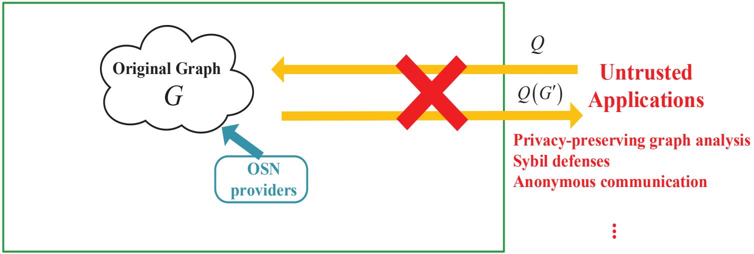
LinkMirage

- LinkMirage Overview
- Algorithm Description
- Privacy Analysis
- Utility Analysis

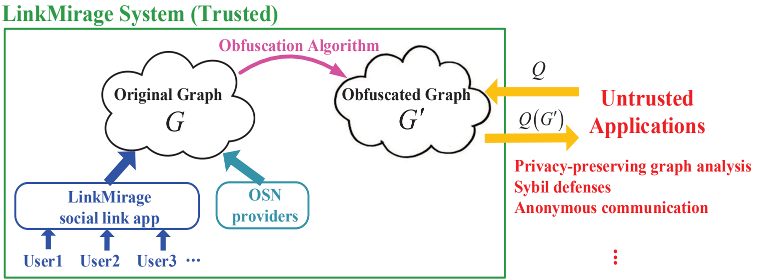
Social Relationship based Applications



Privacy-preserving Social Relationship based Applications



LinkMirage Architecture

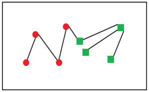


Key intuitions

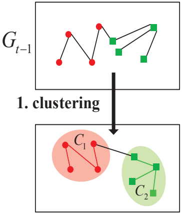
- Naive method: independent perturbation
  - more information is leaked to others
- We need to
  - incorporate graph evolution
  - leverage the information already released in previous graphs

Algorithm Description

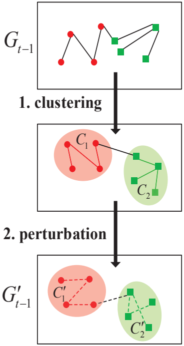
G_{t-1}



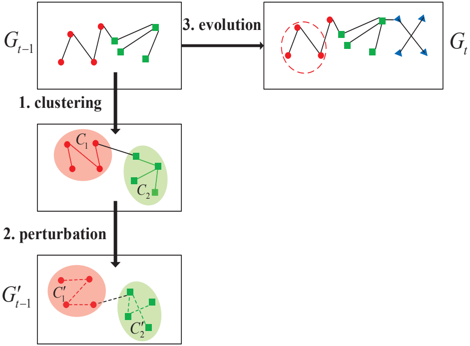
Algorithm Description



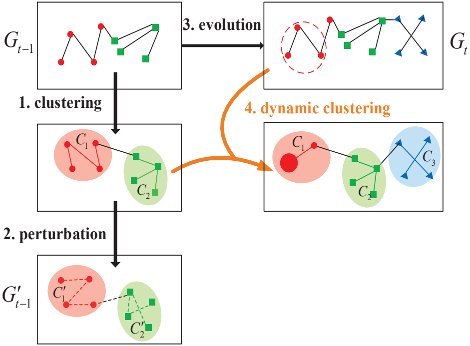
Algorithm Description



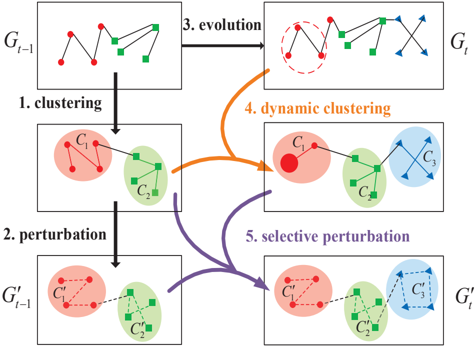
Algorithm Description



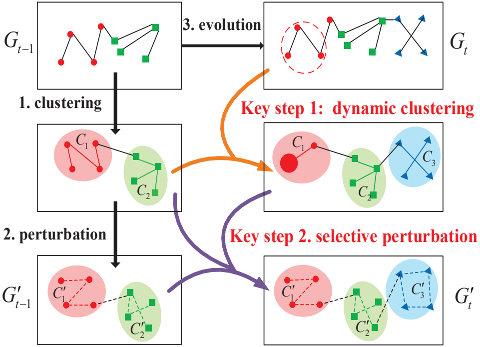
Algorithm Description



Algorithm Description



Algorithm Description

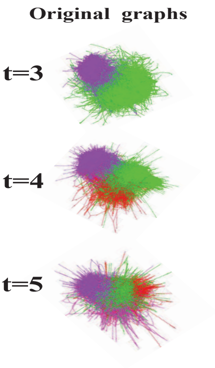


Two Key Steps in Our Algorithm

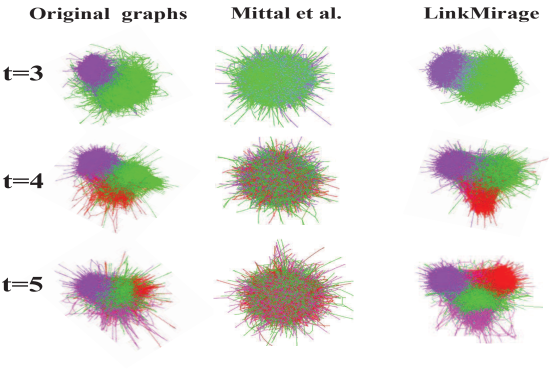
Two key steps

- Dynamic Clustering
  - find communities by simultaneously considering consecutive graphs
  - backtrack based on clustering result of the previous graph
- Selective Perturbation
  - perturb the minimal amount of edges
  - use a very high privacy parameter while preserving structural properties (utility)

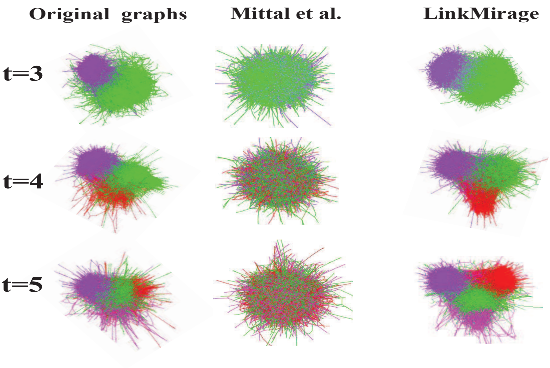
Facebook Temporal Dataset (46,952 users and 876,993 edges)



Utility Advantage

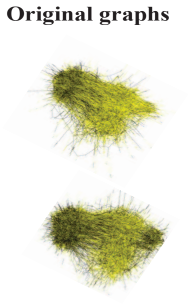


Utility Advantage



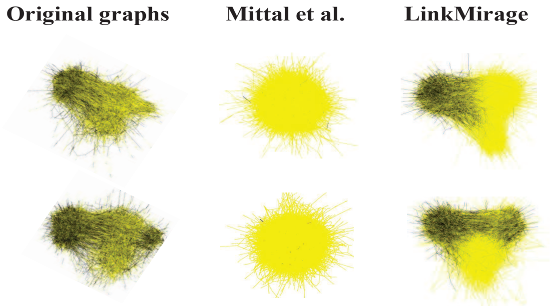
Superior utility, due to dynamic clustering
Utility advantage even exists in static scenario

Privacy Advantage



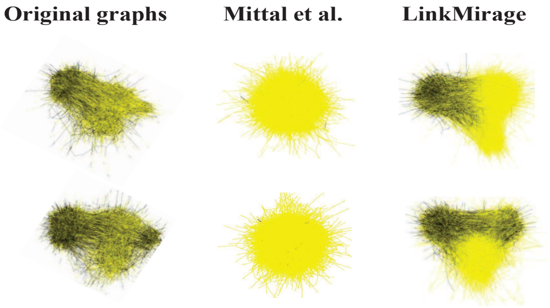
Overlapped edges (black) and Changed edges (yellow) between consecutive graphs

Privacy Advantage



Overlapped edges (black) and Changed edges (yellow) between consecutive graphs

Privacy Advantage



Overlapped edges (black) and Changed edges (yellow) between consecutive graphs

Superior privacy, due to selective perturbation

Anti-Inference Privacy

Assume the worst-case adversary knows

- the obfuscated graphs $\{G'_i\}_{i=0}^t$
- all the other links except for one link L_t
- our obfuscation algorithm

The adversary computes the posterior probability

$$P(L_t|\{G'_i\}_{i=0}^t, W) = \frac{P(\{G'_i\}_{i=0}^t|L_t, W) \times P(L_t|W)}{P(\{G'_i\}_{i=0}^t|W)} \quad (1)$$

and compare with the prior probability

Anti-Inference Privacy

Assume the worst-case adversary knows

- the obfuscated graphs $\{G'_i\}_{i=0}^{t}$
- all the other links except for one link L_t
- our obfuscation algorithm

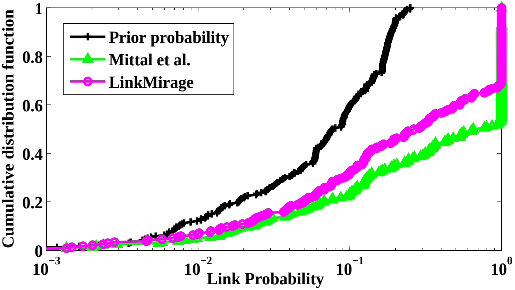
The adversary computes the posterior probability

$$P(L_t|\{G'_i\}_{i=0}^{t}, W) = \frac{P(\{G'_i\}_{i=0}^{t}|L_t, W) \times P(L_t|W)}{P(\{G'_i\}_{i=0}^{t}|W)} \quad (2)$$

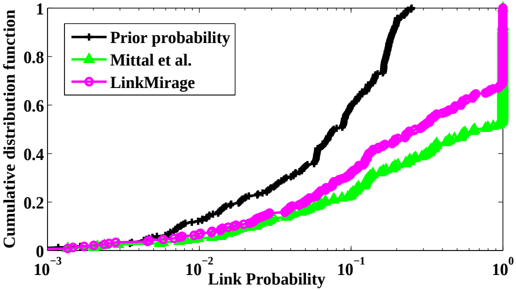
and compare with the prior probability

Higher similarity implies better anti-inference privacy

Anti-Inference Privacy



Anti-Inference Privacy



LinkMirage achieves higher anti-inference privacy!

LinkMirage

- LinkMirage Overview
- Algorithm Description
- Privacy Analysis
- Utility Analysis

Privacy-preserving Graph Analytics

Facebook	Original Graph	LinkMirage	Mittal et al.
Modularity	0.488	0.487	0.415

Privacy-preserving Graph Analytics

Facebook	Original Graph	LinkMirage	Mittal et al.
Modularity	0.488	0.487	0.415

LinkMirage preserves graph analytics better!

Other graph analytics: pagerank, etc.

More applications:

- Sybil defenses
- Anonymous communication

Conclusion

Our LinkMirage system

- Both static and temporal graphs
- Provide rigorous privacy advantages
- Show utility advantages theoretically and using real-world applications
- Generalizable to communication networks and web graphs

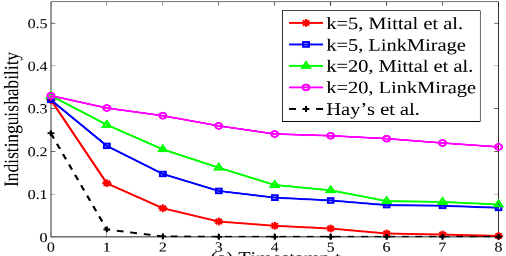
Appendix1: Indistinguishability

Definition

The indistinguishability for a link L_t that the adversary can infer from the perturbed graph G'_t under the adversary's prior information $\{\widetilde{G}_i(L_t)\}_{i=0}^{t}$ is defined as

$$\text{Privacy}_{\text{id}} = H(L_t | \{G'_i\}_{i=0}^{t}, \{\widetilde{G}_i(L_t)\}_{i=0}^{t}) \tag{3}$$

Appendix1:Indistinguishability



(a) Timestamp t

Appendix2:Anti-aggregation Privacy

Definition

The anti-aggregation privacy for a perturbed graph G'_t with respect to the original graph G_t and the perturbation parameter k is

$$\mathrm{Privacy}_{\mathrm{aa}}(G_t, G'_t, k) = \|P_t^k - P'_t\|_{\mathrm{TV}} \tag{4}$$

Appendix2:Anti-aggregation Privacy