PPLP: Privacy-Preserving Link Prediction

Kaleb Kim
Wiam Skakri
Carson Whitehouse
Jonah Lorenzo
Kent Manion
Aidan Bugayong
Case Western Reserve University
Cleveland, Ohio, USA

Abstract

Link prediction on a single graph is well studied; combining multiple graphs (distributed link prediction) can improve accuracy but raises serious privacy concerns. We describe our plan to build a Python library that enables *privacy-preserving link prediction* (PPLP): multiple data holders compute link-prediction scores over their combined graph without revealing their private graph structure. Our approach builds on private set intersection (PSI) and homomorphic encryption, following the protocol of Ayday et al., and we plan to use the ultra-fast PSI of Ling et al. for practical performance. This midterm report states the problem, surveys related work, outlines our planned methods, next steps, and expected deliverables.

Keywords

link prediction, private set intersection, homomorphic encryption, distributed graphs, privacy

1 Problem Statement and Literature Search

1.1 Link Prediction

Link prediction aims to discover unobserved or latent connections between nodes in a graph [4]. Given a snapshot of a network at time t, the goal is to predict which edges will appear by a future time t' using structural information (e.g., proximity measures or graph neural networks). Applications include friend recommendation in social networks, product recommendation in e-commerce, planning in telecommunications, and association discovery in bioinformatics.

Link prediction is typically performed on a single local graph. **Distributed link prediction** considers the setting where two or more parties hold related graphs (e.g., overlapping node sets or the same domain). Merging these graphs can yield more accurate predictions, but sharing raw graph data raises **privacy risks**: identity disclosure, link disclosure, and attribute disclosure. We therefore focus on **privacy-preserving link prediction (PPLP)**: multiple data holders collaboratively compute link-prediction scores over their combined graph *without* revealing their private graph structure to each other.

CSDS 356/456, Cleveland, OH

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/XXXXXXX.XXXXXXX

1.2 Related Work

Link prediction. Liben-Nowell and Kleinberg [4] formalize the link-prediction problem using only network topology (no node attributes) and study proximity-based measures such as common neighbors, Jaccard, and Adamic–Adar.

Privacy-preserving link prediction. Ayday et al. [2] give a two-party protocol in which each party holds a graph; they compute *common neighbors* over the union of both graphs using private set intersection (PSI) and additively homomorphic encryption. The common-neighbor count is expressed as local terms plus crossover terms minus overlaps, computed with a small number of PSI calls. Their threat model assumes semi-honest adversaries and no trusted third party. They also discuss a heavier homomorphic variant to reduce intermediate leakage.

Private set intersection. PSI lets two parties compute the intersection of their sets while revealing only the result. Chen et al. [1] give a high-performance PSI protocol from fully homomorphic encryption (FHE) with communication complexity $O(N_{\text{small}} \cdot \log N_{\text{large}})$. Lattice-based FHE [3] made such protocols practically relevant. Ling et al. [5] propose an ultra-fast PSI from efficient Oblivious Key-Value Stores (OKVS) and Vector Oblivious Linear Evaluation (VOLE), with $O(n)$ communication and very low encoding redundancy (on the order of 1%), which we plan to use for our implementation.

2 Planned Methods

2.1 Link-Prediction Measures

We will support at least one of the following proximity-based measures, which can be expressed using set operations over neighbor sets and thus combined with PSI.

- **Common Neighbors (CN):** For nodes u and v, $\text{CN}(u, v) = |N(u) \cap N(v)|$. More common neighbors imply higher link likelihood.
- **Jaccard:** $\frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$, which accounts for the relative overlap.
- **Adamic–Adar (AA):** $\sum_{w \in N(u) \cap N(v)} \frac{1}{\log |N(w)|}$, emphasizing common neighbors with lower degree.

2.2 Two-Party Common Neighbors via PSI

Following Ayday et al., for two parties Alice (graph G_A) and Bob (graph G_B), the common-neighbor count for a candidate pair (A, E) can be decomposed into:

- Local intersections: $I_A(A, E) = N_A(A) \cap N_A(E)$ and $I_B(A, E) = N_B(A) \cap N_B(E)$ (each party computes locally).
- Crossover terms: e.g., $N_A(A) \setminus I_A$ with $N_B(E) \setminus I_B$, etc., which require PSI so that only intersection sizes (or masked sets) are revealed.

The total $|\mathrm{CN}(A, E)|$ is then obtained as $|I_A| + |I_B| - |I_A \cap I_B|$ plus the contributions from the crossover PSI results (e.g., $|I_{A+B}|$ and the sizes of the privately computed crossover intersections). The protocol uses a small, fixed number of PSI calls per candidate pair.

2.3 Privacy Primitives

- **Private Set Intersection (PSI):** Core primitive to compute intersections (or their sizes) over neighbor sets without revealing the sets themselves. We plan to base our implementation on the ultra-fast PSI of Ling et al. (OKVS + VOLE) for $O(n)$ communication and low bandwidth.
- **Homomorphic encryption (HE):** Used in the protocol to compute on encrypted neighbor sets (e.g., for the polynomial-based PSI of Chen et al. or as in Ayday et al.); FHE allows operations on ciphertexts without decryption.

2.4 System Architecture

- **Core backend (C++):** High-performance PSI and cryptographic primitives (e.g., leveraging an existing implementation such as ShallMate/fastpsi for the Ling et al. protocol).
- **Python bindings:** Pybind11 or Cython to expose the backend to Python, so that data scientists can run PPLP without writing low-level code.
- **Optional front-end:** A web dashboard (e.g., React or Streamlit) for uploading graph snapshots and visualizing predicted links.

3 Next Steps

(1) Integrate or implement the ultra-fast PSI protocol (Ling et al.) and verify correctness and performance in our setting.
(2) Implement the two-party common-neighbor protocol (and optionally Jaccard and Adamic–Adar) using a fixed number of PSI calls per node pair as in Ayday et al.
(3) Design and implement the Python API (graph input, parameterization, and output of link scores or top-k predictions).
(4) Set up multi-machine communication (separate endpoints for each party) so that two instances of the library can run on different computers and execute the PPLP protocol.
(5) Optionally explore the heavier homomorphic variant of Ayday et al. for reduced leakage, and add a simple web UI for demos.

4 Expected Output

We aim to deliver:

- An **integrable Python library** that allows users to run privacy-preserving link prediction on distributed graphs with a clear, documented API.
- **Support for at least the Common Neighbors** measure in the two-party setting, with the option to extend to Jaccard and Adamic–Adar.
- **Example use cases** (e.g., social networks, telecommunications, or e-commerce) demonstrating how to supply graphs and obtain link scores or recommendations without sharing raw graph data.
- Optionally, a **web dashboard** for uploading graphs and visualizing predicted links.

Success is measured by: (1) correctness of the protocol (matching non-private common-neighbor counts where applicable), (2) practical runtime and communication cost, and (3) usability for developers who wish to integrate PPLP into their own applications.

Acknowledgments

This project is conducted for CSDS 356/456. We thank the instructors and the authors of the cited works for the foundations on which we build.

References

[1] Hao Chen, Kim Laine, and Peter Rindal. 2017. Fast Private Set Intersection from Homomorphic Encryption. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. ACM, 1243–1255. doi:10.1145/3133956.3134067

[2] Didem Demirag, Mina Namazi, Erman Ayday, and Jeremy Clark. 2022. Privacy-Preserving Link Prediction. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer International Publishing, 35–50. doi:10.1007/978-3-031-25734-6_3

[3] Craig Gentry. 2009. Fully Homomorphic Encryption Using Ideal Lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing (STOC '09)*. ACM, 169–178.

[4] David Liben-Nowell and Jon Kleinberg. 2007. The Link-Prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology* 58, 7 (2007), 1019–1031. doi:10.1002/asi.20591

[5] Guowei Ling, Hao Chen, Kim Laine, and Peter Rindal. 2025. Ultra-Fast Private Set Intersection From Efficient Oblivious Key-Value Stores. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Implementation: github.com/ShallMate/fastpsi.