Department of Computer Science, University of Helsinki

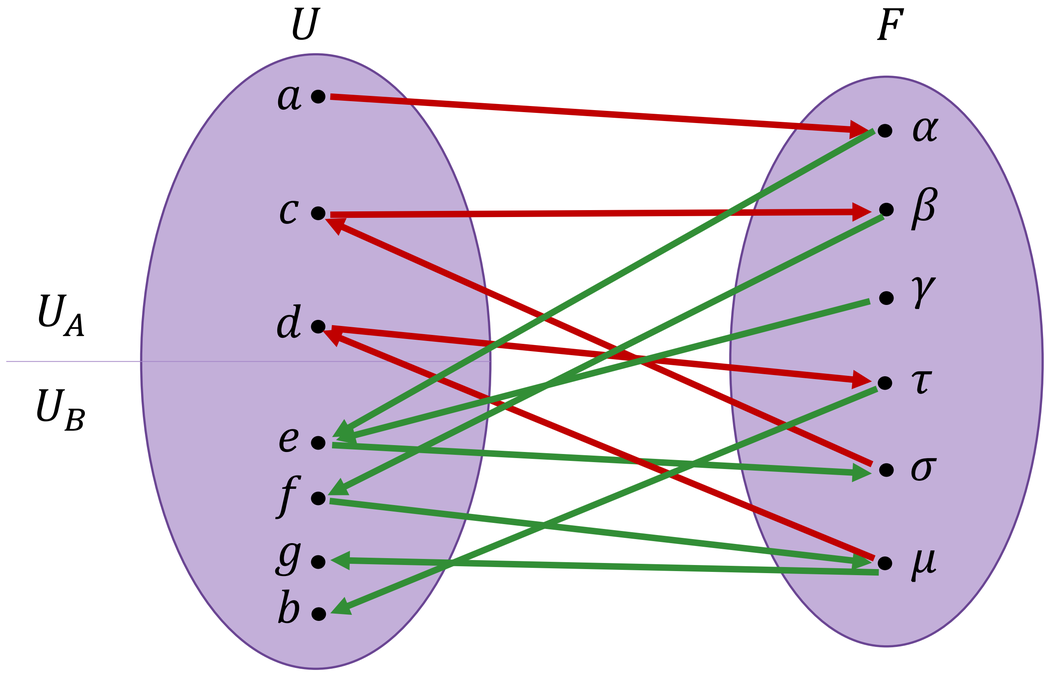
Sara Ramezanian, Tommi Meskanen and Valtteri Niemi
(sara.ramezanian@helsinki.fi, tommi.meskanen@helsinki.fi, valtteri.niemi@helsinki.fi)

Privacy Preserving 2-party Queries on Bipartite Graphs with Private Set Intersection

- **Problem:** The graph has two owners A and B. The goal is to enable owners to find out whether there is a path from node a (belonging to A) to node b (belonging to B). This is done in such a way that each owner does not learn anything else about the other owner's subgraph.

- **Set up:** The database consists of pairs of triplets, that are of form (source-user, source-host, fingerprint) and (fingerprint, target-user, target-host). These define **trust relations** between users on different hosts. This database can be illustrated by a directed bipartite graph $G = (U, F, E)$. The set U consists of (user, host) pairs, and the set F consists of fingerprints.

- **Private Set Intersection (PSI);** is a cryptographic protocol for two parties that both have a set of elements. The goal is to compute the intersection of these two sets, without revealing anything about the elements that are not in the intersection.



E_A: Red arrows
E_B: Green arrows

- **Queries on Bipartite Graphs:** The owners A and B pick a PSI protocol and perform the query as follows.

Owner A

Input: U_A, F_A, E_A and $a \in U_A$

Owner B

Input: U_B, F_B, E_B and $b \in U_B$

First PSI

F_{AB} Output: $F_{AB} = \{f \in F | \exists$ red arrow to f and green arrow from $f\}$ F_{AB}

Second PSI

F_{BA} Output: $F_{BA} = \{f \in F | \exists$ green arrow to f and red arrow from $f\}$ F_{BA}

Finds $F_0^A = \{g \in F_{AB} | \exists$ path from a to g, using only red arrows$\}$. If $F_0^A \neq \emptyset$, sends F_0^A to B. Otherwise, stops (no path).

F_0^A →

If there is an all-green path from some $g \in F_0^A$ to b, stops (path from a to b exists). Otherwise, finds $F_0^B = \{h \in F_{BA} | \exists\ g \in F_0^A$ and an all-green path from g to $h\}$. If $F_0^B \neq \emptyset$, sends F_0^B to A. Otherwise, stops (no path).

Finds $F_1^A = \{g \in F_{AB} \backslash F_0^A | \exists\ h \in F_0^B$ and an all-red path from h to $g\}$. If $F_1^A \neq \emptyset$, sends it to B. Otherwise, stops (no path).

← F_0^B

F_1^A →

If there is an all-green path from some $g \in F_1^A$ to b, stops (path from a to b exists). Otherwise, finds $F_1^B = \{h \in F_{BA} \backslash F_0^B | \exists\ g \in F_1^A$ and an all-green path from g to $h\}$. If $F_1^B \neq \emptyset$, sends F_1^B to A. Otherwise, stops (no path).

← F_1^B

⋮

Repeats until the question is answered

Repeats until the question is answered

- **Conclusion:** We evaluated the performance of our protocol and found it efficient. It only reveals small amount of information about the graph. Future work could study the case where the bipartite graph is shared between more than two parties.

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

ssh communications security

cloSer

BUSINESS FINLAND