Querying a Graph Database You can Trust

FAN Zhe[1]

supervised by

Dr. BYRON CHOI, Koon Kau[1]

joint work with

PENG Yun[1], XU Jianliang[1], HU haibo[1] and Sourav S Bhowmick[2]

Department of Computer Science, Hong Kong Baptist University, Hong Kong[1]

School of Computer Engineering, Nanyang Technological University, Singapore[2]

March 30, 2013

1. Graph Database
   - Category of graph database
   - Category of graph query processing
2. Outsourcing graph database
   - Challenges on managing graph database
   - System model for query services
   - Concerns of employment of query services
3. Authenticated Graph Query Services
   - System overview for authentication
   - Authenticated subgraph query services
4. Privacy-preserving graph query services
   - System overview for privacy preservation
   - Overview of our techniques

Category of graph database

The *graph database* involves:

1. Millions of graphs with modest size, *e.g.*, Protein Database, GENE Database, etc; or
2. One graph with very large size, *e.g.*, Social Network, Road Network, etc.

Category of graph processing

Large amount of different kinds of *graph query processing* are proposed over graph database:

- subgraph (super graph) query processing;
- similarity graph query processing;
- reachability query processing; and
- shortest path/distance query processing;
- ...

Challenges on managing graph database

However, technically challenges of hosting graph database are emerged as

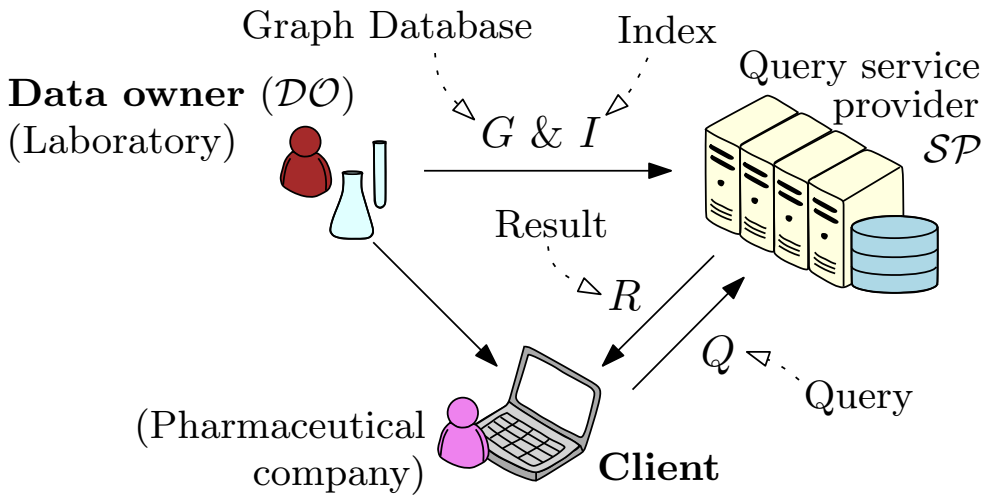
- the unprecedented rate of *increasing volume* of the graph data, *e.g.*, Facebook social graph grows over 800 Millions nodes;
- the complex method of *maintenance* of graph data, *e.g.*, complicated indexing techniques; and
- the high computational *evaluation* of graph queries, *e.g.*, SubIso is NP-hard.

Therefore, the employment of the *query services*, supported by high performance computing (*e.g.*, cloud or clustered computers), have become a practical or even imperative choice.

System model for query services

Three parties in our system model for query services

- *Data owner* ($\mathcal{DO}$), the owner of the graph data and the designer of the graph indexing techniques;
- *Query service provider* ($\mathcal{SP}$), with high computational utility, evaluates the queries from the client on behalf of the $\mathcal{DO}$; and
- *Client* or *User*, issues his/her queries to $\mathcal{SP}$.



Concerns of employment of query services

Followings are two main concerns considered in our research

- What if $\mathcal{SP}$ is *malicious* and *adversary*?
  - $\mathcal{SP}$ may *alter* graph data or the index structure, introduce *wrong* answers, skip certain answers or abort the evaluation.
- What if $\mathcal{SP}$ is *curious*?
  - $\mathcal{SP}$ may be interested in inferring some *private* or confidential information from the graph data or queries to obtain illegal profit.

In this case, how can we **TRUST** those query services?

What if $\mathcal{SP}$ is malicious and adversary?

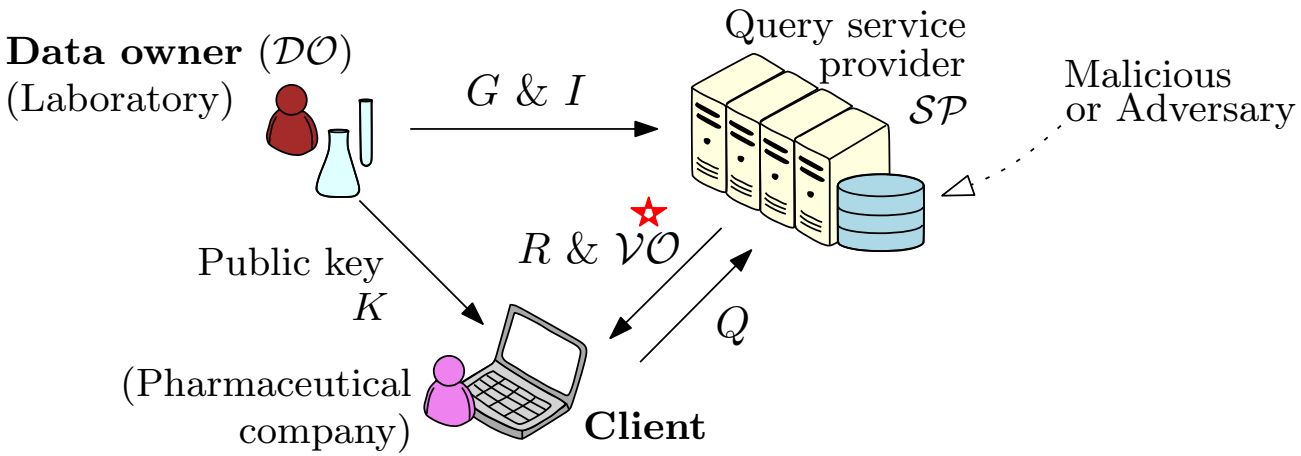
As $\mathcal{SP}$ may be *malicious* or *adversary*, *authenticated graph query services* are in demanded.

In this context, clients or users are able to verify the *authenticity* of the query results, where the authenticity consists of

- *Soundness*: all results are answers and not tampered with; and
- *Completeness*: there is no missing answer in the results.

System overview for authentication

The basic idea for authenticated graph query is to introduce *Verification Object* ($\mathcal{VO}$), which is an auxiliary data structure to store the processing traces such as index traversals.



Therefore, an *efficient authentication* techniques are needed to

- minimize the size of $\mathcal{VO}$; and
- improve the authentication time at client.

System overview for authentication

Different graph query processing results in different authentication techniques.

However, there is very few related work in the literature of graph database. (Authenticated shortest path search [ICDE' 2010], Authenticated graph without leaking [EDBT' 2010])

In our research, we propose two techniques to separately solve

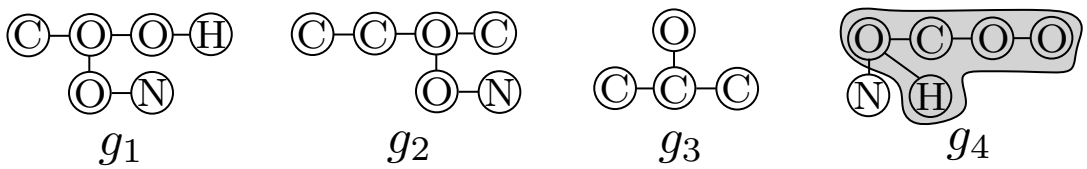
1. authenticated subgraph query services; and
2. authenticated similarity query services.

Authenticated subgraph query services

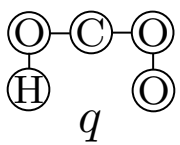
We briefly introduce the *authenticated subgraph query services*.

The graph database we considered is with large amount of graphs with modest size.

The *subgraph query processing* over the graph database can be illustrated as follows.

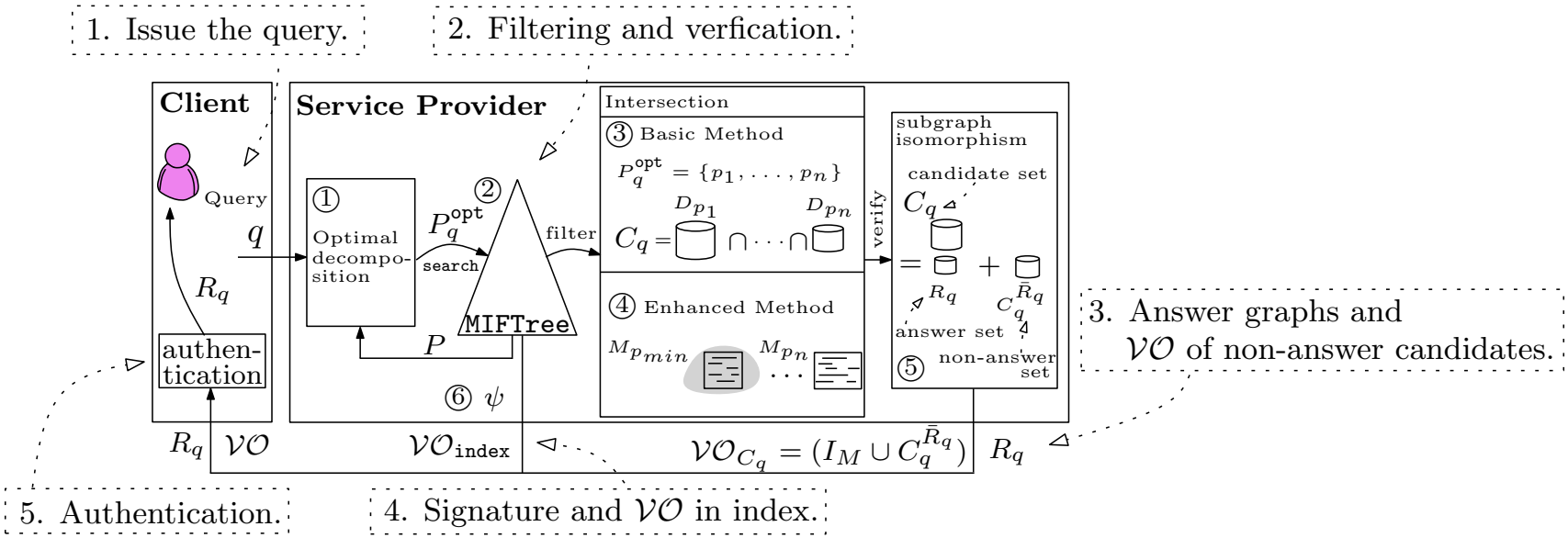


$$G = \{g_1, g_2, g_3, g_4\}$$



Authenticated subgraph query services

The overview of our techniques is illustrated as follows.



- We proposed `MIFTree` for basic authentication;
- We proposed a novel matrix representation of intersection for enhanced authentication;
- We further optimized to cluster "intersect-able" graphs in authentication.

What if $\mathcal{SP}$ is curious?

As $\mathcal{SP}$ may be *curious*, *privacy-preserving graph query services* are needed.

In this context, we concern about the *privacy* of the graph data and graph queries, where the privacy can be the

- size of the graph (number of nodes or edges);
- degree of each node of the graph;
- neighbour information of the graph; or
- structure of the graph;
- ...

System overview for privacy preservation

In our research, we propose to solve the

- privacy-preserving reachability/shortest distance query services;
- privacy-preserving subgraph isomorphism query services.

Our target is to protect the *structure information* of both of the graph data and graph queries.

System overview for privacy preservation

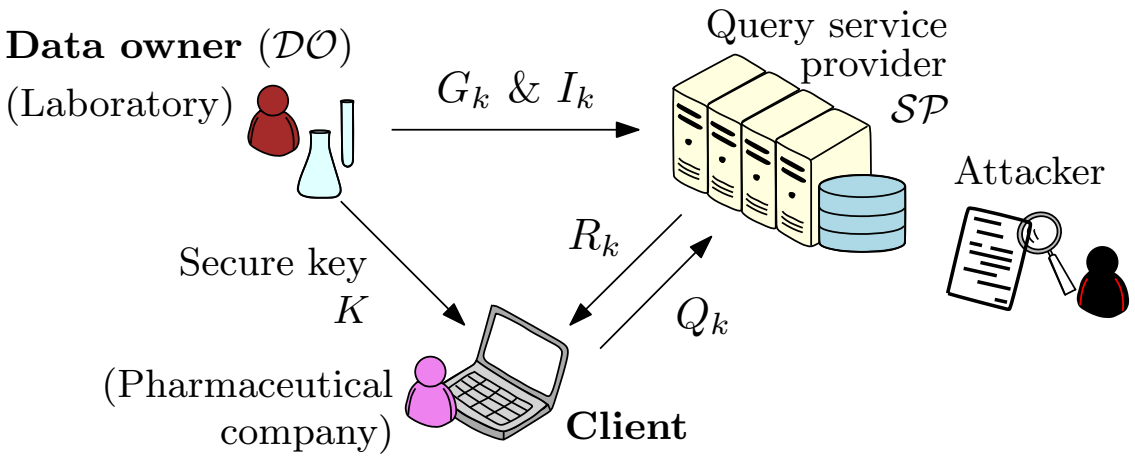
Recently, three techniques are to guarantee the privacy of the graph data in the literature:

- *Graph data obfuscation*, which may introduce some probability for attackers to hack the private information, *e.g.*, shortest distance search [SIGMOD'2011], reachability analysis [WI-IAT'2010];
- *Private Information Retrieval* (PIR), which can guarantee strong privacy, but is with high computational cost $O(\sqrt{N})$, *e.g.*, shortest path search [VLDB'2012], KNN query [VLDB'2010]; and
- *Cryptographic encryption*, which is widely used in traditional database, and spatial database, etc. However, it is **rarely** studied in graph database. Only subgraph query [ICDCS'2011].

Overview of our techniques

1. Encrypt the graph data and index.
2. Outsource the encrypted data to $\mathcal{SP}$.
3. Deliver the secure key to client.

5. Evalute the query processing.



4. Encrypt the query data.
6. Decrypt the encrypted result.

The privacy guarantee is based on the encryption of the graph data and the queries.

This is the end of my presentation.

Questions?