Structure-Preserving Subgraph Query Services
(Extended Abstract)

Zhe Fan†§, Byron Choi†, Qian Chen†, Jianliang Xu†, Haibo Hu† and Sourav S Bhowmick†
†Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR, China.
Email: {zfan, bchoi, qchen, xujl, haibo}@comp.hkbu.edu.hk
§Research and Standard Department, Huawei Software Technologies CO. LTD.
Email: fanzhe@huawei.com
†School of Computer Engineering, Nanyang Technological University, Singapore.
Email: assourav@ntu.edu.sg

I. INTRODUCTION

Subgraph query (via subgraph isomorphism) is a fundamental and powerful query in various real graph applications. It has actively been investigated for performance enhancements recently. However, due to the high complexity of subgraph query, hosting efficient subgraph query services has been a technically challenging task, because the owners of graph data may not always possess the IT expertise to offer such services and hence may outsource to query service providers (SP). SPs are often equipped with high performance computing utilities (e.g., a cloud) that offer better scalability, elasticity and IT management. Unfortunately, as SPs may not always be trusted, security (such as the confidentiality of messages exchanged) has been recognized as one of the critical attributes of Quality of Services (QoS) [4]. This influences the willingness of both data owners and query clients to use SP’s services. Recently, there is a bloom on the research on query processing with privacy preservation, e.g., in the context of relational databases, spatial databases and graph databases. However, up to date, private subgraph query has not yet been studied.

Motivating example: Consider a pharmaceutical company with revenue that depends mostly on the invention of health care products. The company may have discovered new compounds to form other compounds via certain chemical reactions (a structural pattern from the network). On the one hand, the company is reluctant to expose the queries (the interactions) to the SP, as it may apply for patents for the synthesis. On the other hand, the owner of the pathway networks may not only lack the expertise to host query services but may also be reluctant to release the networks to the public. The owner is willing to release it to paid users only. It is crucial to protect both the queries and the network from the SP.

This paper studies that the client may prefer not to expose the structure of query graphs to the SP, and the data owner may not want the SP to be able to infer the structure of their graph data. The problem is elaborated as follows.

System model. We follow the system model that has been well received in the literature of database outsourcing (shown in Fig. 1), and known to be suitable for many applications. It consists of three parties:

1. **Data owner**: He/she owns a database of voluminous data graphs of modest sizes. He/she encrypts each graph and then outsources the encrypted graph to the service provider, and delivers the secret keys to clients for encryption of the query graphs and decryption of the encrypted result;
2. **Service provider (SP)**: The SP may be equipped with powerful computing utilities such as a cloud. The SP evaluates a client’s encrypted query over the encrypted data, on behalf of the data owner, and returns the encrypted result to the client; and
3. **Client**: A client encrypts the query graph Q using the secret keys, submits it to the SP, and decrypts the returned encrypted result to obtain the final answer.

Attack model. We assume the dominating semi-honest adversary model from literature, where the attackers are honest-but-curious and the SP may also be the attacker. For presentation simplicity, we often term the attackers as the SP. We assume that the attackers are the eavesdroppers and adopt the chosen plaintext attack. We assume that the SP and clients are not allowed to collude.

Privacy target. We assume that the privacy target is to protect the structures of a query graph Q and a graph data G from the SP under the attack model defined above. The structural information of Q and G considered is the adjacency matrices of Q and G, respectively. More specifically, the probability that the SP correctly determines the values of the adjacency matrix of the graph is guaranteed to be lower than a threshold with reference to that of random guess.

The problem statement of this paper can be stated as follows: Given the above system and attack model, we seek an efficient approach to facilitate the subgraph isomorphism query services with preserving the above defined privacy target.

---

†In addition to privacy protection via legal means, this stream of research has aimed to offer technological solutions for such protection.
To our knowledge, such a problem has never been addressed before. The intrinsic difficulty of this research is that the $SP$ cannot optimize query processing by directly using the structures of the graph, since such information cannot be exposed. However, most of the existing subgraph isomorphism algorithms (e.g., VF2 [1], QuickSI [5] and Turboiso [3]) for the query services must traverse the graph, which by definition leaks structural information. A naive method is to transfer the entire database to the client for query processing. However, it is inefficient when the database is large.

II. OVERVIEW OF OUR PROPOSED SOLUTIONS

Our techniques for a structure-preserving subls (denoted as $SP_{subls}$) [2] are derived from the Ullmann’s algorithm [6], a seminal algorithm for subgraph isomorphism. We revise the Ullmann’s algorithm into three steps that form the foundation of our techniques. (1) Enumerates all possible subgraph isomorphism mappings $M_I$’s from query graph $Q$ to data graph $G$; (2) Match verifies if the mapping $M_I$ is valid or not; and (3) Refine reduces the search space of $M_I$’s by degree and neighborhood constraints. The benefits of adopting the Ullmann’s algorithm are twofold: (1) the query evaluation between $Q$ and $G$ is mostly a series of matrix operations between their adjacency matrices $M_Q$ and $M_G$. It does not require traversals on structures; and (2) its evaluation requires simple structures. This makes the privacy analysis simpler.

We transform the above three steps into a series of matrix computations, denoted as $Tsubls$. $Tsubls$ comprises three steps: (1) $TEnum$ enumerates all $M_I$’s; (2) $TMatch$ verifies the validity of $M_I$ by additions and multiplications using $M_Q$ and $M_G$, where $M_G$ is the complement of $M_G$; and (3) $TRefine$ reduces the search space of $M_I$’s by inner products on our proposed static indexes $SI_Q$ and $SI_G$ of $Q$ and $G$, where $SI_Q$ ($SI_G$) is an ensemble of $h$-hop information of each vertex of $Q$ ($SI_G$) represented by a bit vector.

$SP_{subls}$ is then proposed on top of $Tsubls$. The overview of our techniques is presented in Fig. 2. We first propose a new private-key encryption scheme, namely cyclic group based encryption scheme (CGBE). The data owner transforms, indexes, and encrypts a data graph $G$ into $G_k$ offline. The encrypted graph $G_k$ is outsourced to the $SP$. The client obtains a secret key from the data owner and encrypts his/her query $Q$ as $Q_k$. He/she submits the encrypted query $Q_k$ to the $SP$.

1. We develop $SP_{Refine}$ which exploits private inner products on the static indexes ($SI_Q$ and $SI_G$) to derive a refinement that reduces the number of possible mappings $M$ between $G_k$ and $Q_k$. The static indexes of the graphs were computed and encrypted off-line, whereas those of the queries are computed once by the clients online. We analyze the effects of these optimizations on the probabilities that the $SP$ may correctly determine graph structures. Therefore, the clients may tune the trade-off between refinement performances and privacy requirements.

2. We propose $SP_{Enum}$ which optimizes the search for mappings by introducing a protocol that involves the client’s participation, who informs the $SP$ useless mappings. This provides $SP$ some structural information and thus we quantify its effect on privacy.

REFERENCES


