1. Motivation

Many modern big-data applications work on top of graph-structured datasets. For instance, graph databases such as Neo4j, TitanDB, OrientDB, and others handle large-scale graphs that model biological interactions and social networks.

Accessing network-structured data from/to secondary storage must ensure access locality for graphs.

In multi-tenant environments, protecting sensitive information from unauthorized access and leakage is imperative. Unfortunately, although existing solutions for the graph processing model include secure communication and authentication (for independent graphs and authenticated entities); there is no solution for multiple authenticated entities (data owners) sharing a diversity of data graphs.

2. Threat Model

We assume a probabilistic polynomial time adversary. Leakage-free is when the user cannot infer any information from the subgraph $G_i$ that belongs to $G$ but no $G_j$.

Inference attack: A user $U_j$ with access to one or more subgraphs $G_i \subseteq G$, attempts to infer sensitive information from the signature and $G_i$ (e.g., node or edge inference).

Data tampering attack: An attacker tampers the structural order, the content, the relation between two nodes (edges), or a set of blocks $B \in \beta$.

Data-processing services involve mutually distrustful parties.

3. Our Contributions

The definition of the practical and formal security model of authentication for the blocked graph.

Three new algorithms for

- block construction with high locality
- disk layout for authenticated and blocked graphs
- unique oblivious data structures and programming API for graph access utilizing trusted hardware [4]

An open source implementation and evaluation of the proposed techniques (AB-Graph) [2].

4. Authentication Graph and Layout Ordering (Proposed Solution)

5. Implementation


We use METIS partitioning framework to compare against our implementation (experimental results).

JAVA API with Java Native Interface to call native algorithms.

5.1. AB-Graph: A Peer-to-Peer Based Graph Datastore

5.2. AB-Graph: An Authenticated Blocked Graph (Proposed Solution)

5.3. AB-Graph: Inference attack

5.4. AB-Graph: Data tampering attack

5.5. AB-Graph: Leakage-free

5.6. AB-Graph: Inference attack

5.7. AB-Graph: Data tampering attack

5.8. AB-Graph: Leakage-free

5.9. AB-Graph: Inference attack

5.10. AB-Graph: Data tampering attack

6. Applications


2. Authenticated and Private Network Sharing for enhanced Graph Analytics. Our project provides novel approaches for Analyzing Biological Networks.

3. Authenticated and Private Smart Farming.

4. Data Leakage Detection on Social Networks.

5. Authenticated Peer-to-Peer Distributed Live Media Streaming with WebRTC [1].

7. References


