Privacy-Preserving Computations on MapReduce

Shlomi Dolev, Yin Li, Shantanu Sharma
Department of Computer Science, Ben-Gurion University of the Negev, Israel
{dolev,yin,sharmas}@cs.bgu.ac.il*

Data and computation outsourcing move databases and computations from a private and trusted cloud to a public cloud, which is not under the control of a single user. Thus, the outsourcing results in less burden on a private cloud in terms of the maintenance of databases, infrastructures, and executions of queries. Unfortunately, the ease in storing data and executing computations in the public clouds implies a risk of violating security and privacy of the databases and the computations.

MapReduce [1] was introduced by Google in 2004. MapReduce provides efficient and fault tolerant parallel processing of large-scale data without dealing with security and privacy of data and computations. While MapReduce is not directly related to the public clouds, many public clouds, *e.g.*, Amazon Elastic MapReduce, Google AppEngine, IBM Blue Cloud, and Microsoft Azure, enable users to perform MapReduce cloud computations without considering physical infrastructures and software installations. Thus, the deployment of MapReduce on the public clouds enables users to process large-scale data in a cost-effective manner and establishes a relation between two independent entities, *i.e.*, the public clouds and MapReduce.

An example of Secure and privacy-preserving equijoin of two relations $X(A,B)$ and $Y(B,C)$. *Problem statement*: The join of relations $X(A,B)$ and $Y(B,C)$, where the joining attribute is B, provides output tuples $\langle a,b,c\rangle$, where (a,b) is in A and (b,c) is in C. In the equijoin of $X(A,B)$ and $Y(B,C)$, all tuples of both the relations with an identical value of the attribute B should appear together for providing the final output tuples.

Consider that the relations X and Y belong to two organizations, *e.g.*, a company and a hospital, while a third user wants to perform the equijoin. However, both the organizations want to provide results while maintaining the privacy of their databases, *i.e.*, without revealing the whole database to other organization and the user. Hence, it is must to perform the equijoin in a secure and privacy-preserving manner.

Problem statement. The main obstacle for providing privacy-preserving framework for MapReduce in the adversarial (public) clouds is computational and storage efficiency. An adversarial cloud may breach the privacy of data and computations. Hence, we are interested in making a secure and privacy-preserving computation execution and storage-efficient technique for MapReduce computations in the clouds. We are looking at information-theoretically secure data and computation outsourcing and query execution using MapReduce. Specifically, our focus is on four types of privacy-preserving queries, as follows: `count`, `search` and `fetch`, `equijoin`, and fetch tuples with a value belonging in a `range`. By developing privacy-preserving data and computation outsourcing techniques, a user receives only the desired result without knowing the whole database; moreover, the clouds are also unable to know the database and the query.

Our contributions. In this paper, we provide the following:

Information-theoretically secure data outsourcing. We provide an information-theoretically secure data and computation outsourcing technique that prevents a malicious cloud provider to know the database and a query. Specifically, we use Shamir secret-sharing [3] for making secret-shares of each tuple of a relation and send them to the clouds. A user can execute her queries using accumulating-automata (AA) [2] on these secret-shares without revealing queries/data to the cloud.

Privacy-preserving query execution by third-parties. We can perform the following operations in a privacy-preserving manner, as: `count`, `equijoin`, `search`, and `fetch`. The main idea is that if we can perform privacy-preserving string matching operations on a database, then using the string matching operations we can perform all the above mentioned operations in a privacy-preserving manner.

Advantages of the proposed approach. The proposed approach has three main advantages, as follows: (*i*) the approach is well suited to the (public) cloud environment; (*ii*) eliminates the need of a database owner in terms of the database maintenance and query processing, except creating and distributing secret-shares to the clouds; and (*iii*) the query response time is also smaller than the response time of a query over an encrypted database.

1. REFERENCES

[1] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *OSDI*, pages 137–150. USENIX Association, 2004.

[2] S. Dolev, N. Gilboa, and X. Li. Accumulating automata and cascaded equations automata for communicationless information theoretically secure multi-party computation: Extended abstract. In *Proceedings of the 3rd International Workshop on Security in Cloud Computing*, pages 21–29, 2015.

[3] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.