Privacy Preserving Query Processing in the Cloud

Sakina Mahboubi, Reza Akbarinia, Patrick Valduriez

Privacy Preserving Query Processing in the Cloud

Sakina Mahboubi*
Ph.D. thesis start date: Oct. 2015
Supervisers: Reza Akbarinia & Patrick Valduriez
Zenith team, Inria and LIRMM, University of Montpellier, France
sakina.mahboubi@inria.fr

1. INTRODUCTION

Nowadays, cloud data outsourcing provides users and companies with powerful capabilities to store and process their data in third-party data centers. However, when a user stores her sensitive data in a public cloud, they becomes vulnerable to several attacks, e.g., from the employees of the cloud provider.

One solution for protecting the user data against attacks is to encrypt the data before sending them to the cloud servers. Then, the challenge is to answer user queries over encrypted data. A naive solution for answering queries is to retrieve the encrypted database from the cloud to the client, decrypt it, and then evaluate the query over *plaintext (non encrypted)* data. This solution is not practical, in particular for large databases.

In this PhD thesis, we are interested in processing top-k queries on encrypted data. These queries have attracted much attention in several areas of information technology such as sensor networks [15], stream management systems [13, 12] and spatial data analysis [1, 3]. A top-k query allows the user to specify a number k, and the system returns the k tuples which are most relevant to the query. The relevance degree of tuples to the query is determined by a *scoring function*.

There have been many different approaches proposed for processing top-k queries over plaintext data. Two of the best known approaches are FA [5] and TA [7] that work on sorted lists of attribute values. These approaches, particularly TA, can find efficiently the top-k results because of smart strategies for deciding when to stop reading the database. However, TA, FA and all other efficient top-k approaches developed so far assume that the data are in plaintext, and there is no efficient solution capable of evaluating efficiently top-k queries over encrypted databases.

When we think about top-k query processing on encrypted data, the first idea that comes to mind is the utilization of a fully homomorphic encryption cryptosystem, e.g. [8], which allows to do arithmetic operations over encrypted data. this type of encryption allows to compute the overall score of data items over encrypted data. However, fully homomorphic encryption methods are very expensive in terms of encryption and decryption time. In addition, they do not allow to compare the encrypted data, and to find the top-k results.

We proposed efficient approaches, called EncFA and BuckTop, for processing top-k queries over encrypted data. We evaluated their response time over encrypted data with that of the TA algorithm over original (plaintext) data. Our results show that the response time of BuckTop over encrypted data is close to TA over plaintext data, and even better over some large databases.

In the rest of this paper, we fist define the problem we address. Then, we briefly introduce our proposed approach, and then we discuss the related work.

2. PROBLEM DEFINITION

The problem which we address is top-k query processing over encrypted data. Let D be a database, and $e(D)$ be its encrypted version such that each data $c \in e(D)$ is the ciphertext of a data $d \in D$, i.e. $c = Enc(d)$ where $Enc()$ is an encryption function. The database $e(D)$ is stored in a remote server.

Given a number k and a scoring function f, our goal is to develop an algorithm A, such that when A is executed over the database $e(D)$, its output contains the ciphertexts of the top-k results, i.e. those that can be found by executing a correct top-k algorithm over the database D.

3. APPROACH

The architecture of our system for query processing over encrypted data is composed of two parts: *Trusted client* and *Service provider*.

Trusted client is responsible of user data encryption and decryption, user access control and secure key management. When a query is issued by a user, the trusted client checks the access rights of the user. If the user does not have the required rights to see the query results, then her demand is rejected. Otherwise, the issued query is transformed to a query that can be executed over the encrypted data and is sent to the service provider. the Trusted client decrypts the received results (calculated by the service provider) and returns to the user the k data item which are the response of the query launched.

Service provider stores the encrypted data, executes on them the algorithms provided by the trusted client, and returns the results to it.

We propose two approaches for processing top-k queries over encrypted data. The first approach, called *EncFA*, uses

BDA 2016, 15 au 18 Novembre, Poitiers, France.

a probabilistic (semantic) encryption scheme to encrypt the attribute values of the database items. It also uses a deterministic scheme for encrypting the IDs of data items. Then, it sends the encrypted database to the service provider. the data items after encryption must have the same order that they had before being encrypted.

The second approach, called *BuckTop*, is more efficient than EncFA. It uses a bucketization technique to partition the data items in each list into a set of buckets. also it uses tow types of encryption schema to encrypt the database; one is deterministic used to encrypt data item IDs and the other is probabilistic used to enrypt the attribute values of the data items. BuckTop includes a top-k query processing algorithm that works on the encrypted data of the buckets, and returns a set containing the top-k results. It also includes an efficient filtering algorithm that filters the false positives as much as possible in the server. We prove theoretically the correctness of the BuckTop approach.

4. RELATED WORK

A first important paper in top-k query processing is [5], which models the general problem of answering top-k queries using databases organized into lists of data items sorted by their local scores. One of the most efficient algorithms over sorted lists is the TA algorithm, which was proposed by several groups [6, 9]. However, all these algorithms assume that the data scores are available as plaintext, and not encrypted.

There have been also some proposed solutions for secure kNN query processing, e.g. [4, 2, 16]. The problem is to find k points in the space that are the nearest to a given point. This problem should not be confused with the top-k problem in which the given scoring function plays an important role, such that on the same database and with the same k, if the user changes the scoring function, then the output may change. Thus, the proposed solutions proposed for kNN cannot deal with the top-k problem.

The bucketization technique has been used in the literature for answering range queries over encrypted data, e.g. [10, 11] where Hore et al. use this technique, and propose optimal solutions for distributing the encrypted data of a database to the buckets in order to guarantee a good performance by reducing the number of false positives while preserving a high security level. The techniques developed in [10, 11] can be used in our system for an optimal distribution of the encrypted data in the buckets.

The only paper which we found about top-k query processing over encrypted data is [14] published in arXiv.org. The proposed architecture assumes the existence of two non-colluding servers s_1 and s_2 in two different clouds. One of the servers, say s_2, has the decryption keys, and the other one, say s_1, stores the data. Top-k query processing proceeds by using the TA algorithm and accessing the encrypted data in s_1, such that after reading each data in s_1, its encrypted local scores are sent to the server s_2 (using a special protocol) where they are decrypted and compared with the TA threshold. Our assumptions about the cloud are different. In our solution, we do not need to trust on any remote server, and during the top-k query processing, we do not decrypt the encrypted data in the cloud servers. In addition, the solution in [14] needs a lot of communications between remote servers (i.e., at least two messages after each sorted access). This solution that is not efficient and incurs a high latency in the query processing time.

to the best of our knowledge, in the literature there is no efficient solution for processing top-k queries over encrypted data. In this work, we propose such a solution.

5. REFERENCES

[1] L. Chen, J. Xu, X. Lin, C. S. Jensen, and H. Hu. Answering why-not spatial keyword top-k queries via keyword adaption. In *ICDE Conf.*, pages 697–708, 2016.

[2] S. Choi, G. Ghinita, H. Lim, and E. Bertino. Secure knn query processing in untrusted cloud environments. *IEEE Trans. Knowl. Data Eng. (TKDE)*, 26(11):2818–2831, 2014.

[3] F. M. Choudhury, J. S. Culpepper, and T. K. Sellis. Batch processing of top-k spatial-textual queries. In *Second International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data*, pages 7–12, 2015.

[4] Y. Elmehdwi, B. K. Samanthula, and W. Jiang. Secure k-nearest neighbor query over encrypted data in outsourced environments. In *ICDE Conf.*, pages 664–675, 2014.

[5] R. Fagin. Combining fuzzy information from multiple systems. *J. Comput. Syst. Sci.*, 58(1):83–99, 1999.

[6] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *PODS Conf.*, 2001.

[7] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. *J. Comput. Syst. Sci.*, 66(4):614–656, 2003.

[8] C. Gentry. Fully homomorphic encryption using ideal lattices. In *ACM Symposium on Theory of Computing (STOC)*, pages 169–178, 2009.

[9] U. Güntzer, W. Balke, and W. Kießling. Towards efficient multi-feature queries in heterogeneous environments. In *2001 International Symposium on Information Technology (ITCC)*, pages 622–628, 2001.

[10] B. Hore, S. Mehrotra, M. Canim, and M. Kantarcioglu. Secure multidimensional range queries over outsourced data. *VLDB J.*, 21(3):333–358, 2012.

[11] C. Li, M. Hay, G. Miklau, and Y. Wang. A data- and workload-aware query answering algorithm for range queries under differential privacy. *PVLDB*, 7(5):341–352, 2014.

[12] Z. Shen, M. A. Cheema, X. Lin, W. Zhang, and H. Wang. Efficiently monitoring top-k pairs over sliding windows. In *ICDE Conf.*, pages 798–809, 2012.

[13] X. Wang, Y. Zhang, W. Zhang, X. Lin, and Z. Huang. SKYPE: top-k spatial-keyword publish/subscribe over sliding window. *PVLDB*, 9(7):588–599, 2016.

[14] H. Z. Xianrui Meng and G. Kollios. Declarative cleaning of inconsistencies in information extraction. *arXiv:1510.05175v2*, 2016.

[15] H. Yang, C. Chung, and M. H. Kim. An efficient top-k query processing framework in mobile sensor networks. *Data Knowl. Eng.*, 102:78–95, 2016.

[16] B. Yao, F. Li, and X. Xiao. Secure nearest neighbor revisited. In *ICDE Conf.*, pages 733–744, 2013.