International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072

Top-k Query Processing Using Top Order Preserving Encryption (TOPE)

Ajisha K Sivan

Student, Dept. of Computer Science Engineering, Thejus Engineering College, Kerala, India

---

Abstract - *With the dramatic increase on the scale of datasets, the management of dataset became more difficult. To reduce both local storage and query processing overhead the data owner can upload the data to the public cloud. But the public cloud is semi trusted, it may reveal our private information. Privacy of outsourced data is a major concern. Currently there exist several methods to ensure privacy of outsourced data. Top-k queries can retrieve most relevant k tuples from the huge dataset. In this paper we propose a extended Top Order preserving Encryption (TOPE) which supports top-k queries. The mutable TOPE it supports only top-1 queries. So here we use an extension of mutable TOPE. The specialty of Order Preserving Encryption (OPE) is that it allows comparison operations directly on the encrypted data. In TOPE the order of plaintext remains same in the cipher text domain. This method ensures the privacy of our outsourced data.*

Key Words: **Top Order Preserving Encryption, Order Preserving Encryption, mutable TOPE, query processing, top-k queries, etc…**

1. INTRODUCTION

Top-k queries can retrieve the most relevant k tuples from the huge datasets. Currently there exist several methods for top-k query processing. The methods return k tuples as a result of query processing. . With the dramatic increase on the scale of datasets, a growing trend is for data owners to outsource their large scale datasets to public cloud services in order to reduce local storage and query processing overhead. Due to legal and commercial issues, privacy of outsourced datasets on the cloud side is still a major concern. For example, an inside attacker who can see all cloud side data can easily reveal the sensitive data if an outsourced dataset is stored in plaintext [1]. To ensure privacy we can use different methods. One such method is traditional encryption. It is nothing but encrypting the owners dataset before outsourcing to the cloud. But it losses search functionalities on the cloud.

In this paper we propose a Top Order Preserving Encryption (TOPE). It naturally answers most relevant 1 tuple from the dataset. Here we extend TOPE to support top-k queries with minimum privacy leakage. This method provides more privacy than OPE. In the current existing systems have drawbacks such as they are time consuming, generate performance overheads and do not ensure the privacy. The paper mainly focuses on the privacy of data. The method helps to provide top-k query results from the encrypted data with minimum privacy leakage. Also helps to increase the search performance.

1.1 Order Preserving Encryption

The Order Preserving Symmetric Encryption (OPE) is a deterministic encryption scheme whose encryption function preserves numerical ordering of the plaintexts. OPE scheme allows comparison operations to be directly applied on encrypted data, without decrypting the operands. Thus, equality and range queries as well as the MAX, MIN, and COUNT queries can be directly processed over encrypted data. Similarly, GROUP BY and ORDER BY operations can also be applied. Only when applying SUM or AVG to a group do the values need to be decrypted[2]. The OPE can answer top-k queries correctly and naturally.

The TOPE is mutable, means cipher text of some data may change when new cipher texts are computed. In TOPE order of plaintext remains same in ciphertext domain. Generally OPEs are interactive between the client and server. The top-k queries are applied to numeric values.

- I implement this scheme based on a binary heap with Java and use JDBC (Java Database Connectivity) to connect to MySQL to query datasets. Encrypted data are searched with SQL queries.

2. RELATED WORKS

OPE was first defined by N. Chenette et al. [3]. They proposed ideal security of OPE. But they failed to achieve complete privacy because half of the plaintext bits were leaked. *Vasilis Pappas et al. [4]* designed a private DBMS, named Blind Seer which is capable of scaling to tens of TB's of data. In this method identifies small privacy-critical Sub problems in a large problem and solve those securely. Then use the outputs of the subtasks to complete the large task efficiently. The method solves the problem by traversing an encrypted search tree. *Jaideep Vaidya et al. [5]* proposed a secure method for doing top-k selection from vertically partitioned data. They proposed a vertically partitioned data distribution model.

Mohamed A. Soliman et al. [6] the paper introduces a new probabilistic formulation for top-k queries. The formulations are based on traditional top-k semantics and possible world's semantics. This method is the first one to address top–k query processing under possible worlds semantics. Formulated the problem as a state space search, and introduced several query processing algorithms. Which optimality guarantees on the number of accessed tuples and materialized search states. *Yanchao Zhang et al. [7]* paper considers a novel distributed system for collaborative location-based information generation and sharing which become increasingly popular due to the explosive growth of Internet-capable and location-aware mobile devices. The system consists of a data collector, data contributors, location-based service providers (LBSPs), and system users. The data collector gathers reviews about points-of-interest (POIs) from data contributors, while LBSPs purchase POI data sets from the data collector and allow users to perform location-based top-k queries.

F. Kerschbaum et al. [8] presents an ideal-secure, but significantly more efficient order preserving encryption scheme. The scheme work based on average height of random binary search trees. The input to the encryption algorithm is a plaintext. Encryption is stateful and stores an ordered list of plaintext- ciphertext pairs. The ciphertext is sent to the database server. The encryption algorithm is keyless. The state of the algorithm plays the role of the key. The size of the state of the encryption algorithm is the size of the dictionary of the database. The update algorithm potentially updates all ciphertexts produced so far. It re-encrypts all (distinct) plaintexts in order, i.e. the median element first and so on. Thus, it produces a (temporarily) balanced tree. The state of the encryption algorithm is updated on the database client. This updated state needs to be sent to the database server and its persistent data needs to be updated potentially all database rows. This affects not only the column store, but also the entire dictionary.

3. METHODOLOGY

With the increase on the size of datasets, the data owners outsource their large scale datasets to public cloud. This helps to reduce both local storage and query processing overhead. But cloud is semi trusted. So privacy of outsourced data is a major problem. Taking into account various methods identified from the literature, a framework model for top-k query processing is proposed.

3.1 System Model

Fig – 1 shows the system model, there have 3 entities they are Data Owner, Data User and Server. Data owners and data users must register in the system before entering to the system. Both user and owner maintains same key for encryption or decryption. The data owner has huge dataset which have to be outsourced into the public cloud to reduce both local storage and query processing overheads. Initially the data owner has a secret key to perform encryption operation. To maintain the order of plaintext, the data owner performs min order and max order sorting for each attribute(For example: If there have 2 attributes age and salary then owner generates min and max list of age, and min and max list of salary).Performs encryption in the sorted list using the secret key. After encryption the owner uploads the data to the public cloud. Whenever the cloud receives the ciphertext then it will generate heap for both min sorted and max sorted list of each attribute to store the sorted data.

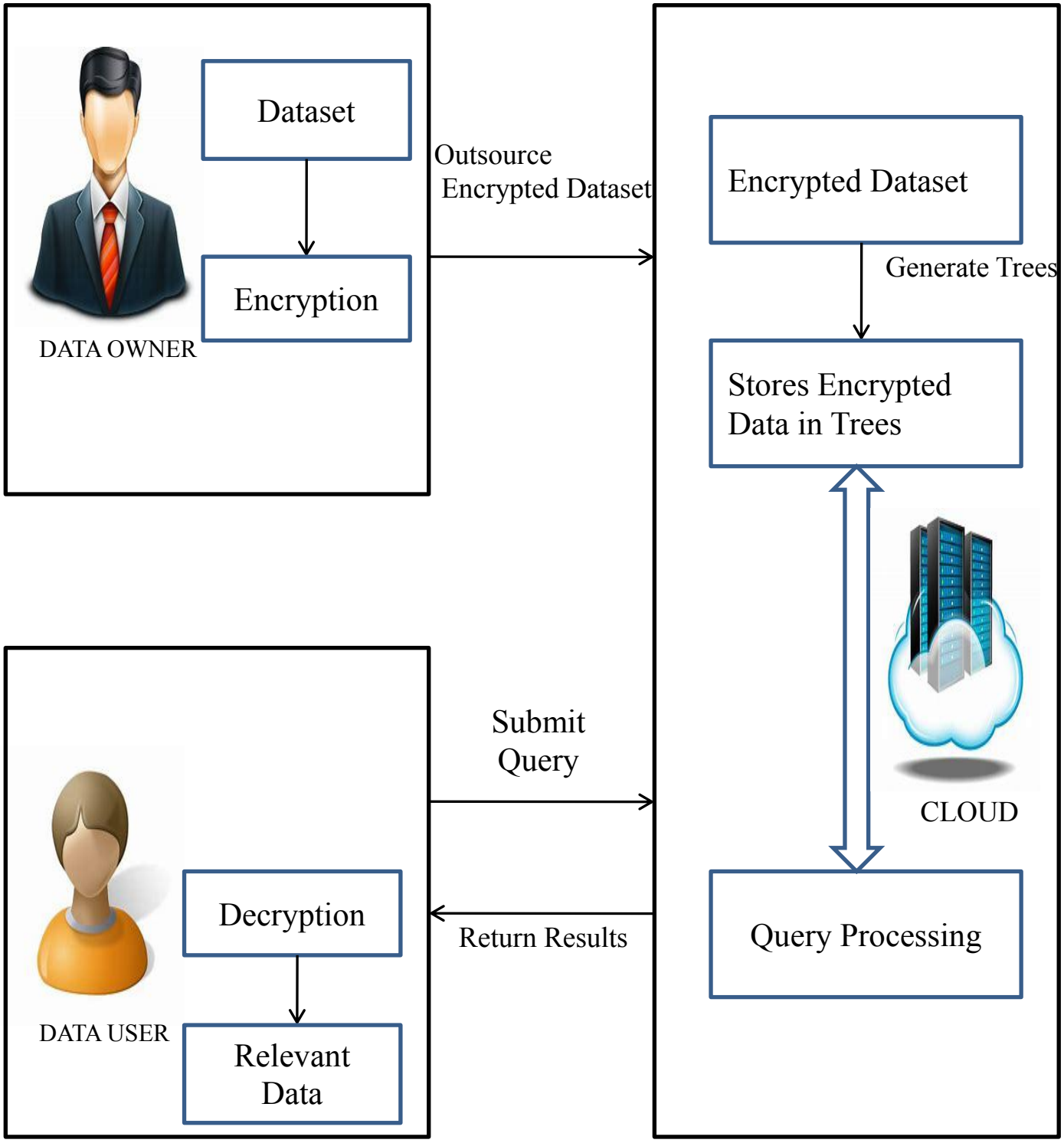

Fig -1: System Model

The data user has a secret key for decryption. User can submit queries such as MIN, MAX and COUNT. When user submits query then the server will select the corresponding tree based on the type of query ie; if it is a MIN query then server will select the MIN heap of specified attribute. If it is a MIN query then the Query includes 2 other fields. One is the attribute type and other one is the value of k (ie; the number of tuples which have to be retrieved). During query processing the server selects the heap based on the user submitted query. After selection he will retrieve the tuples.

If it is a MIN query the server selects the min heap of corresponding attribute and then returns the data stored in the k nodes from top. For COUNT query the user submits attribute and value of attribute. In this case the data owner encrypts the value of attribute and then sends the data to the cloud. The cloud performs search operation in the heap, if the value is in the heap then returns the number of tuples which has the corresponding attribute value. For min and max query cloud returns top k relevant data that is in encrypted form. After receiving the result the user decrypts the data to get the original plaintext of the encrypted data.

4. CONCLUSION

Propose an extended Top Order-Preserving Encryption to enable top-k queries over encrypted data. This method utilizes the partially-order property of heaps which helps to balance the privacy and search functionality. This method helps to protect data from leakage of top-k queries compared to the use of OPE. Top-k queries can retrieve the most relevant k tuples from the huge datasets. The Order-preserving encryption (OPE) can be used for answering top-k queries on encrypted data correctly and naturally. But OPE unnecessarily leaks too much information. The paper proposed a mutable top OPE to overcome the limitation of OPE. The mutable TOPE firstly enable top-1 (min or max) queries on encrypted data with minimized data leakage. This project extends this TOPE to support top-k queries in general. With TOPE, the ciphertexts of top-k values remains top-k in the ciphertext domain. This method utilizes the partial-order property of heaps to Balance the privacy and search functionality. Encrypted data are stored in MySQL database and are searched with SQL queries. This method helps to retrieve most relevant k data from the dataset also helps to find the count of data in the dataset correctly with minimum privacy leakage.

REFERENCES

Hanyu quan and boyang wang, “Efficient and Secure Top-k Queries With Top Order-Preserving Encryption”, in IEEE, June 7, 2018.

Order Preserving Encryption. Accessed: August 29, 2018. [Online]. Available: http://cryptowiki.net/index.php?title=Order-preserving_encryption.

A. Boldyreva, N. Chenette, Y. Lee, and A. O’Neil, ‘‘Order-preserving symmetric encryption,’’ in Proc. EUROCRYPT, 2009, pp. 224–241.

V. Pappas, ‘‘Blind seer: A scalable private DBMS,’’ in Proc. IEEE SP, May 2014, pp. 359–374.

Chris Clifton and Jaideep Vaidya, “Privacy-Preserving Top-k Queries”, International Conference on Data Engineering 1084-4627/05 , 2005

M. A. Soliman, I. F. Ilyas, and K. C.-C. Chang, ‘‘Top-k query processing in uncertain databases,’’ in Proc. IEEE ICDE, Apr. 2007, pp. 896–905.

Rui Zhang and Yanchao Zhang, “Secure Top-k Query Processing via Untrusted Location-based Service Providers”, 2012

F. Kerschbaum and A. Schropfer, ‘‘Optimal average-complexity ideal security order-preserving encryption,’’ in Proc. ACM CCS, 2014, pp. 275–286.