



RANGE BASED QUERIES OVER ORDER PRESERVING ENCRYPTED DATA

VASANTH S (191411044)

UG student

vasanthsgc@gmail.com

Mrs.S.ALAGUMANI

SUPERVISOR

Asst. Professor

Department of Computer Science and Engineering

Saveetha School of Engineering, Thandalam, Chennai-602105

Tamil Nadu

ABSTRACT:-

The canonical application of OPE is to support efficient range queries over encrypted data, which is relevant in cloud computing and outsourced database management. Since encryption preserves order, a range of cipher texts corresponds directly to a range of plaintexts, and range queries specified by encryptions of the two ends of the range are easily answered by returning every cipher text between those ends. The goal of this project is to provide OPE constructions along with appropriate security definitions. This paper proposes the ordered bucketization technique. Ordered Bucketization is helpful to communicate the range queries over encrypted data without decrypting the encrypted content by attaching a bucket number tag to each block of cipher text. Till now no research has been done for the security of ordered bucketization in a cryptographic sense. This paper proposes an encryption method with Ordered Bucketization and proposes a new security technique for EOB, IND-OCPA-P, which assumes an adversary has sensible power. At last, an Ordered bucketization construction, in which the Encrypted Ordered Bucketization implementation is well secured on the indistinguishability under ordered chosen-plaintext attack model, is proposed. A bucket ID is attached to each divided block of query in ascending order. With relevance the potency of a range query, the projected OB guarantees fairly sensible efficiency on range queries by showing that the distribution of a bucket size isn't skew.

KEYWORDS:-Bucketization, Order Preserving Encryption, Range queries.

INTRODUCTION:

Bucketization is a data partitioning step which provides grouping of records by considering their sensitive data, followed by attaching the random (index) tag to each bucket effectively making every data within a bucket without indistinguishable from one another. When the query is searched by the client (data admin) it is first checked buckets which are intersecting the query using a index number, which is stored on the client (this is only a small amount of data) and all data of the intersecting buckets are extracted from the server. With the use of bucketization and OB,

many types of SQL queries can be sent over the encrypted data are possible, only if the bucket tag, which indicates the original plaintext before encryption, is tagged to each block of encrypted data. For eg, if a client program needs to retrieve the data in the range between 50 to 100, it first determines the total numbers of buckets whose union is the least set that intersect the queried range. The client system sends the bucket tag numbers to the server (database). The server (database) search all the data which has been encrypted and compares its the bucket number with the received numbers. Then the server, returns the

proper data to the client. The client will get the correct result by filtering the data which are not in the range after decrypting the data. This paper helps to retrieve data from outsourced database or datasets. These database or datasets contains encrypted data for security and privacy purpose. Our proposed bucketization scheme is helpful to retrieve data from that encrypted datasets and also providing the encrypted query construction for data retrieval. Bucketization method divide the whole data into several buckets with unique bucket numbers. The user query will fetch to the corresponding bucket number in the datasets and retrieve the data in plaintext format without any data loss. This is the main scope of bucketization technique with Encrypted Ordered Bucketization (EOB) algorithm.

Existing system focuses the range queries and a bucketization based technique to support them in the all the Database As Service models. In the bucketization method, an attribute domain is divided into a cluster of buckets each of which is identified by a number tag. These bucket numbers are kept as an index (referred as the crypto-index) and are used by the server to process all the queries. The system doesn't consider the security issues arising from the creation of bucketization-based index to support the range queries.

- Existing system doesn't consider the security and privacy issues during encryption phase.
- OPE does not support a weak IND-OCPA.

The problem in querying the encrypted data is primarily driven by data outsourcing applications. during a typical setup the service supplier provides the data to its clients and provides

a range of data management functionalities, like queries, updates, modifications etc. While the server is expected to implement the various operations correctly, the client may not trust the server side with complete access to its sensitive data. The solution is to store the sensitive data in encrypted form and keep the keys secret from the server at all times. For querying that encrypted data anyway, the server should be able to produce various query predicates in oppose the data without needing decrypting it.

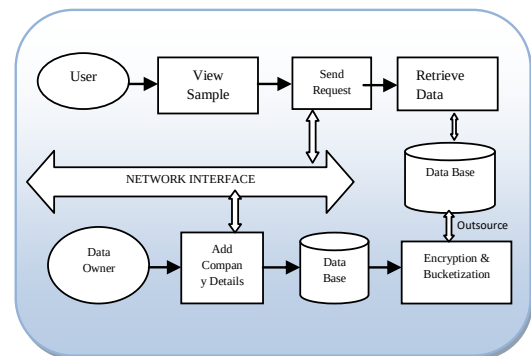
METHODOLOGIES

This paper presents a new encryption method with Ordered Bucketization (EOB) which can be constructed with any OB method. In Encrypted Ordered Bucketization, the output of encryption is in the form of (*bucket#*) *JJ* (*cipher text*), In which the *bucket#* refers to the bucket number tags of messages which are encrypted; and the bucket number is the output of Ordered Bucketization, and the *cipher text* part is the result of the simple symmetric encryption.

SYSTEM DESIGN

The users or nodes involved in this system are Sender, Intermediate and Receiver. In order to send file, the sender has to find out the list of nodes which are connected with the sender. From that available list he can choose receiver. Then the sender has to analyze the performance of each and every node which is connected with the sender. The performance analysis list will return the priority based result so that sender can choose the intermediate to send the file. The Intermediate will receive the file from sender then it will analyze the performance so that it can send data to another intermediate or receiver. In the receiver side, the receiver has to select the file path to receive the file

from sender or intermediate. Then the receiver can view the file received file.



TECHNIQUE USED:

ENCRYPTION WITH ORDERED BUCKETIZATION (EOB).

This paper presents a new method of symmetric encryption with Ordered Bucketization (EOB) which can be constructed by any OB method. In Encrypted Ordered Bucketization, the output of encryption is in the form of $(bucket\#) \text{ } JJ \text{ (cipher text)}$, In which the $bucket\#$ refers to the bucket number tags of messages which are encrypted; and the bucket number is the output of Ordered Bucketization, and the *cipher text* part is the result of the simple symmetric encryption.

STEP 1: The user queries are converted into range queries over the encrypted data in the database or dataset.

STEP 2: The user requested data are retrieving from database in encrypted format and it divided into several sizes of buckets.

STEP 3: Each bucket have a unique bucket numbers for converting cipher text into plain text.

STEP 4: EOB detect and ignore the false positives that are retrieved with cipher text content.

STEP 5: Finally the data are decrypted into plain text format to the user.

CONCLUSION:

The proposed bucketization approach is exceptionally helpful to secure the outsourced database. This approach utilizes the ordered bucket distribution to encrypt the database information and furthermore utilizing the symmetric key algorithm, for example, TripleDES for encryption, it productively encrypt the database tables and decrypting it. The bucket numbers are ordered so the data admin gets the least and large bucket number for decrypt the data instead of getting all bucket numbers. An Experimental outcomes demonstrate the efficient performance of our approach.

REFERENCES

1. *Difference of order statistics in a sample of uniform random variables.* available at: <http://math.stackexchange.com/questions/68749/difference-of-order-statistic-in-a-sample-of-uniform-random-variables>.
2. *OPEN SSL: Cryptography and SSR/TLR Toolkit.* Available at: <https://www.openssl.org>.
3. R. Agarwal, J. Kieman, R. Srikanth, and Y. Xu. Orderpreserving encryption for numerical data. In *SIGMOD Conference*, pages 563–574, 2004.
4. M. Bellare and P. Rogaway. *Course Notes in Introduction to Modern Cryptography.* available

at:<http://cseweb.ucsd.edu/>

mihir/cse207/classnotes.html.

5. A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Orderpreserving symmetric encryption. In *Eurocrypt*, volume 549 of *Lecture Notes in Computer Science*, pages 224–231. Springer, 2010.

6. A. Boldyreva, N. Chenette, and A. O'Neill. Order-preserving encryption revisit: Improved security analytics and alternative solution. In *CRYPTO*, volume 6831 of *Lecture Notes in Computer Science*, pages 578–588. Springer, 2012.

7. D. Bonehn and B. Waters. Conjunctive, subsets, and range queries over encrypted data. In *the Theory of Cryptography Conference (TCC)*, pages 535–551. Springer-Verlag, 2007.

8. Y. Dinhg and K. Kleimn. Model-driven application-level encryptions for privacy of e-health data. In *IEEE International Conference on Availability, Reliability and Security*, pages 341–346, 2010.

9. O. Goldreich. *Foundation of Cryptography*. 2000.

10. Arjun K. Gupta. *Handbook for Beta Distribution and Its Applications*. CRC Press, 2004.

11. H. N. Nagaraja H. A. David. *Order Statistics (3rd Edition)*. Wiley, New Jersey, 2003.

12. H. Hacigumun, C. Iyer, C. Li, and S. Mehrotra. Execution of sql over the encrypted data in the database-service-provider system. In *ACM SIGMOD02*, pages 216–227. ACM.

13. B. Shore, S. Mehrotra and M. Kantarcioglu. Secured multidimensional range queries over outsourced data. *The VLDB Journal*, 21:333–357, 2011.

14. B. Shore and G. Tsudik. A privacy-preserving index for ranged queries. In *30th VLDB conference*, pages 720–731, 2004.

15. Paul Jaccard. ture comparative de la distribution floral dans une portion des alpes et des jurah. *Bulletin de la Socit Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

