International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 4 (2015) pp. 10405-10413
© Research India Publications
http://www.ripublication.com

Implementing Query Builder In Encryption With Ordered Bucketization (Eob)

Meichingthuan Pamei
Department of Computer Science, Sathyabama University,
Chennai,
chingpamei@yahoo.co.in
Ramesh Kumar Mandal
Department of Computer Science, Sathyabama University,
Chennai,
rameshdeep421@gmail.com.
A.Pravin
Department of Computer Science, Sathyabama University,
Chennai,
pravin_ane@rediffmail.com.

Abstract

To implement the query builder, ordered bucketization is an approach to be followed. It is a cryptographic object. It concentrates more on security features including sharing of data. This paper describes and encryption schemes with ordered bucketization. While in ordered bucketization plaintext space are divided into several disjoint bucket number from 1 to n. Each are ordered in ascending order. It provides better efficiency in queries with security features. In this proposed ordered bucketization, (n-1) points selected and are equally distributed. It is based on selected points.

Index Terms/keywords: Bucketization, Data encryption, Data integrity, Cryptography.

Introduction

In the proposed Ordered Bucketization (OB), the plaintext are divided in many disjoints buckets number ranging from 1-n.These bucketization method divide the whole data into several buckets with unique bucket numbers. In OB many types of queries can be encrypted multiple times if the bucket number that corresponds to the existing plaintext before encryption is attached to each encrypted data.

In a situation when the client wants to retrieves data from the server database, it will circulates the number of buckets with the least queries range and send it to the database server. The database server will search the bucket number through all the encrypted data. Once the server found the desire data, it is then sends back to the client. On receiving the data from the server, the client willdecrypt the received data and filter out unwanted data to obtain desire information.

The above given situation are overcome in the proposed system with ordered bucketization. Whereclient will receive only the require data. In the existing technique order-preserving encryption (OPE), there is some inefficiency which can be resolved by implementing encryption with ordered bucketization (EOB).

Order-preserving encryption model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which extremely reduces execution costs. The client will keep at constant the set of data about data to verify the proof. In OPE techniques the range of queries encryption are limited. OPE doesn't support weaker version of IND-OCPA (INDistinguishability under Ordered Chosen Plaintext Attack) security.

The existing system has inefficiency in security measures. The measure of lack of security is the major hindrance in the existing system. And in order-preserving techniques we can generates an encryption key for a limited time. It is only one time key generation method.

Encryption is usually performed for a particular bulk of data frames that contains collections of data.

Encryption with Ordered Bucketization (EOB) is a new symmetric encryption techniques defined in OB. These encryption form can be given as in (bucket)/(ciphertext). Bucket represents the number of buckets of messages encrypted which signifies the OB result. And ciphertext represents the result of symmetric encryption.

Problem Existing

The problem in which system doesn't consider the privacy threats arising from the creation of bucketization-based indices to support range queries is study in this paper. With bucketization and the EOB construction various range ofqueries encryption are defined amd made it possible. Bucketizationallowbucketing of number of buckets from the data and then encrypt the data. This number of buckets obtain from the database are assigned with a unique number to each bucket. We can discuss this paper by taking the query operation in which the client encrypted the data and sends request to the server. The encrypted data assigned into bucketand concatenated with the ciphertext with specific range. The server will use the bucket number to execute the queries operation. Let's take an example on how the bucketization helps in executing the plaintext efficiently in sending the request from the client system and retrieval of the necessary data from the server in secure manner. The original plaintext is divided into multiple buckets (1-n). This can be illustrate as 1, 2, 3, 4, …n numbers of buckets with the plaintext space (0,n). This will be allotted in random permutation as shown.

Bucketization approaches for multiple number of buckets:

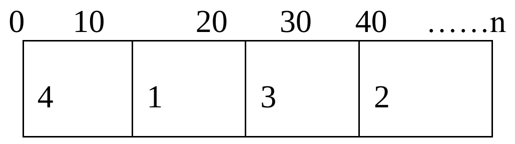

Figure 1 (i): Random permutation bucketing function (0-n)

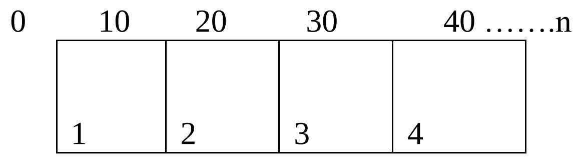

Figure: 1(ii): Order-preserving bucketing (0-n)

The above bucketing approach shows bucketing functions from [0, n] in the both cases. The first approach used permutation for random number of buckets, while in the later follows permutation of order-preserving approach. In table one the client request range of query to the server in random manner where the server responds to the requested query and send the required data to the client. The server will process the range of query and sends back the results to the users. This deals with the possible bucketization method in querying and in security measures.

In random permutation when the client sends the query request to the server, the bucket numbers is allotted with a plaintext each say (5, n) that includes 1, 2, 3, 4…n. On receiving the request from the user in an encrypted format the server will compute the encrypted data and check the bucket numbers where they are assigned and send the results to the client. Here the server will check all the data in which bucket number they are assigned. In the receivers end the client will perform post processing to see whether the retrieved data is correct or not. With the help of decryptions process the user will decrypt all received data in the range of query and then remove the unwanted messages that are included in the bucket. The range of query the client received say 5 and n may contains false positives. Whereas data within the range of query 10 and 20 i.e., [1, 3] as in table contains the data user request absence false positive. By performing post processing in the client side the valued data information is retrieve after removing the unwanted range of query. In query optimized bucketing system the false positive are reduced. The experiment that covers the greatest number of the altered lines is given most astounding need, and executed first [17].

When the query request is made by the client in case of order-preserving bucketing, a minimum number of buckets with the maximum rate bucket number consisting of plaintext will be send. The range of query is always in and order manner in this type. As in the table given the bucket numbers including plaintext ranging from (5, n) with 1, 2, 3, 4….n will be fetched to the server system in an encrypted format. The server will accept the requested range of query from the client and compute them to generate the information. The bucket number in the range of [1, 2, 3, 4] will be send back to the client in order to find whether the data has a maximum value or equal values to 1, or has minimum value or equal values to 4. Here the number of false positive are increased and the bucket number are decreased.

Analysis of Existing System

The existing techniques focus on encryption of block of buckets using order-preserving encryption. Initially client will encrypts all the data set using order preserving encryption and keep in the server database. Client will encrypt data for a selected range of query at the minimum number and the maximum number and generate a single key. The generated key is used by the client during encryption of data in the server system. Once the data is retrieve from server in encrypted format, client will decrypt the data received and get the correct message absence false positive value. Even though there is less occurrence of false positive in OPE system, this cannot be perform for higher range of queries resulting to inefficiency in security measures. Since OPE does not support the weaker version of IND-OCPA security model. The OPE method do not provide a secure and efficient query results to the users because this system does not consider threats to users privacy. Therefore this is not IND-OCPA-P secure because it will not support any version of the security model. Hence the existing system does not deal with the security features.

In the existing Order-Preserving Encryption (OPE) techniques the key generation is limited to one time and it is usually done for a single block of bucket containing set of data at a time. Encryption is done in terms of bucket that contains data set. The OPE techniques generates probabilistic proofs of possession by sampling random sets of blocks from the server. The led to reduce in execution costs, but the user has to keep the amount of set of data about data to verify the results provided.

Analysis of Proposed System:

The proposed system used the Encryption with Ordered Bucketization (EOB) techniques with the Ordered bucketization (OB) where the security features and its efficiency are overlooked and deals to provide a secure model. In EOB the encryption are carried out for each data in the bucket in regards to the desired rows or columns. Bucketization allows multidimensional range of query to be encrypted. In the proposed system any version of IND-OCPA security model is support. So the level security of the encrypted data is studied and provides secure manner for the client. Consequently it turns into a key to minimize the test suite and pick a subset of experiments from test suite which will be executed in slightest time and has the ability to cover all the shortcomings. Thus reordering the experiment on the premise of time issue [16].

Techniques Used

The Encryption with Ordered Bucketization (EOB) techniques can generates encryption keys for each data within the selected bucket. A bucket consists of a set of data collection in rows and columns. Encryption is performed for a particular rows and columns. The result of encryption in EOB is in the form of bucket and ciphertext. Bucket represents the number of buckets of messages encrypted which signifies the OB result. And ciphertext represents the result of symmetric encryption.

1. The user queries are converted into range of queries over the encrypted data in the database and sends request to the server database.
2. The retrieval data from the database are in encrypted format. The data are didvide into several buckets sizes and assigned with a unique bucket numbers and convert the ciphertext into plaintext.
3. The ciphertext does not contain false positives, but if detected ignored by the EOB. On the receiving end the encrypted data are then decrypted into plaintext to obtain results.

Encryption order bucketization aims at increasing the efficiency rate. This proposed method has no false positive value during data sending and receiving encryption process inspite of the increased in efficiency.The data is less corrupted during the encryption and decryption processes between the client and server database. The system ought to be equipped for discovering the best set of streamlining systems for any project that is given as info. Since there are more number of systems, discovering the best set is not plausible physically. The procedure of choice is mechanized to minimize the execution time, arrangement time, tuning time and standardized tuning time [18]. Since multiple keys are generated each time the client encrypts and decrypts the data. The data to be encrypted from the bucket are randomly selected and queried.

Diagrammatical Representation

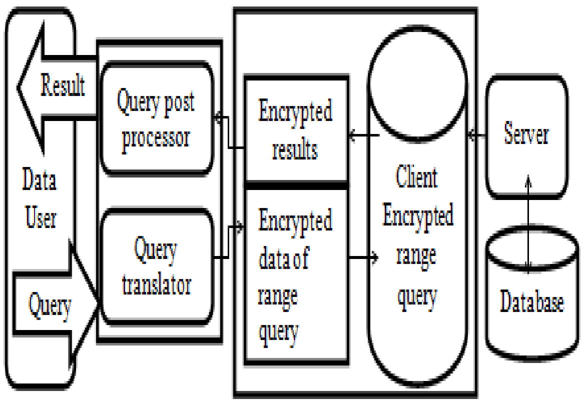

Figure 2: The overall processing of data between the user and server database

Data Owner:-Authentication

In this part of modules the data owner must provide the correct username and password in the login page available. On giving the username and password the system will check in the database whether the given inputs are right or wrong. If the login is successful then it will proceed to the next page that is main page. Otherwise it will remain in the login page itself if the inputs are wrong. It allows only the authenticate users to log in.

Encryption& Bucketization

The data owner will encrypt the company details of the employees and stored in the database. In this phase the data owner will select the columns and rows in the tables and perform encryption. The data columns and rows are assigned with a unique bucket number before encrypting the columns and rows. This will help decryption of the selected rows and columns in the future.

Handle User Request:
This will enable the data owner to handles the data user request. The client will send the bucket number to the data owner for decrypting the particular data column in the database. Then the data owner will views the request and sends the key to decrypt the data in the database table for the particular data user.

Log Management:
Here the data owner will views all the bucket details and its corresponding encrypted table column details. In this module the data owner can delete unwanted buckets from the database and can also reconstruct the existing tables in desire form.

Data User:-

Registration:
In order for the new user to login into the application, first it must register by providing necessary details of the user into the database. Once the new user has completed his sign up process, by using correct username and password the user can login into the application.

Login:
The user has to fetch correct username and password which was provided during the time of registration in order to login. If input is wrong the user will not be able to enter into the application. If the login is success the user will move in to the next page that is main page.

View Sample Data:
In this module the user will be enabled to view the sample of data tables that is provided by the administrator. The tables seen is the encrypted multiple records and multiple subsets that are present in the database.

Send Request:
The data user will select necessary table from the lists of tables that are made available by the data owner. This module allows the data user to select table and view

the encrypted columns of the table and also get the bucket number for the encrypted columns. Then the data user will send buckets number to the data user for decrypting the column values from the database.

View Decrypted Data:
Here the data user will gets the key from the data owner for decrypting the data column. Using this key the user will shows the original decrypted data from the encrypted database.

Results & Description

Graphical representation between Existing and Proposed system

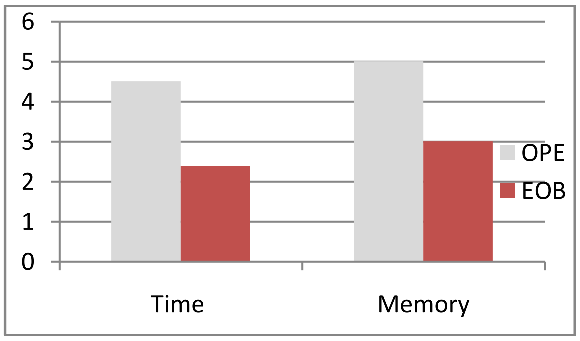

Figure 3 (i): Time and memory is reduced in EOB, while OPE has increased usage of time and memory.

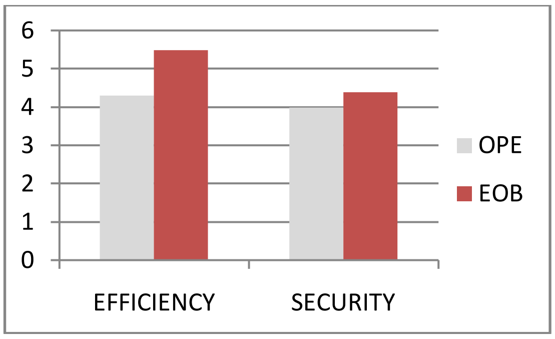

Figure 3 (ii): Better efficiency and security in EOB than in OPE.

Since EOB techniques corroborate with any version of IND-OCPA-P security model, it provides better secured measures. Encryption of the sensitive data from the table is done through stored procedure. In stored procedure everything is done in the database in a more secured way.

Finally the outputs will be viewed in Graphical Users Interface (GUI) of the selected column. It will automatically generate the query for the encrypted data using the key provided by the data owner. The output will be displayed in a tabular format.

Conclusion

An experiment was conducted to overcome the problem in the efficiency and security over the range of query to encryption data, by implementing the Encryption with Ordered Buctization (EOB). In the experiment we found out that by using the encryption with ordered bucketization the security level is drastically improved and the efficiency to support range of query is effectively increased. The proposed EOB system with Ordered Bucketization (OB) was successfully implemented on any IND-OCPA-P security model. From our experiment a drastical improvement has been observed.

References

[1] *Difference of order statistics in a sample of uniform random variables*, available at:http://math.stackexchange.com/questions/68749/differenceof-order-statistics-in-a-sample-of-uniform-random-variables.

[2] *OPEN SSL: Cryptography and SSL/TLS Toolkit*. Available at: http://www.openssl.org.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order-preserving encryption for numeric data, In *SIGMOD Conference*, pages 563–574, 2004.

[4] M. Bellare and P. Rogaway. *Course Notes on Introduction to Modern Cryptography*, available at:http://cseweb.ucsd.edu/ mihir/cse207/ classnotes.html.

[5] A. Boldyreva, N. Chenette, Y. Lee, and A. ONeill. Order-preserving symmetric encryption. In *Eurocrypt*, volume 5479 of *Lecture Notes in Computer Science*, pages 224–241. Springer, 2009.

[6] A. Boldyreva, N. Chenette, and A. ONeill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *CRYPTO*, volume 6841 of *Lecture Notes inComputer Science*, pages 578–595. Springer, 2011.

[7] D. Boneh and B. Waters. Conjunctive, subset, range queries on encrypted data. In *the Theory of CryptographyConference (TCC)*, pages 535–554. Springer-Verlag, 2007.

[8] Y. Ding and K. Klein. Model-driven application-level encryption for the privacy of e-health data. In *IEEE InternationalConference on Availability, Reliability and Security*, pages 341– 346, 2010.

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

[10] Arjun K. Gupta. *Handbook of 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. Hacigumus, B. Iyer, C. Li, and S. Mehrotra. Executing sql over encrypted data in the database-service-provider model. In *ACM SIGMOD02*, pages 216–227. ACM.
[13] B. Hore, S. Mehrotra, M. Canim, and M. Kantarcioglu. Secure multidimensional range queries over outsourced data. *TheVLDB Journal*, 21:333–358, 2011.
[14] B. Hore, S. Mehrotra, and G. Tsudik. A privacy-preserving index for range queries. In *30th VLDB conference*, pages 720– 731, 2004.
[15] Paul Jaccard tude comparative de la distribution florale dans une portion.
[16] Pravin, A. and S. Srinivasan, 2013. Effective test case selection and prioritization in regression testing. J. Comput. Sci., 9: 654-659.
[17] T. Prem Jacob, T. Ravi. Optimal Regression Test Case Prioritization using genetic algorithm.Life Sci J 2013;10(3):1021-1033] (ISSN:1097-8135). http://www.lifesciencesite.com. 149
[18] J.ANDREWS, T.SASIKALA, Efficient framework architecture for improved tuning time and normalized tuning time, WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, 2013; 10(7):230-240 (ISSN:2224-3402). http://www.wseas.org.