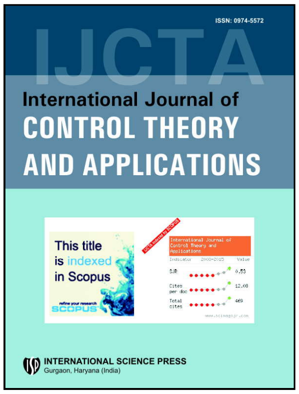


ISSN: 0974-5572

IJCTA

International Journal of CONTROL THEORY AND APPLICATIONS

This title is indexed in Scopus

INTERNATIONAL SCIENCE PRESS
Gurgaon, Haryana (India)

International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 19 • 2017

Response Time Analysis for Querying Data on Order Preserved Encrypted Buckets

Arjun K.R.[1], Praveen K.[1], Santhya R.[1] and Sridhar K.S.[1]

[1] *TIFAC-CORE in Cyber Security, Amrita School of Engineering Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, India, Emails- arjun92kr@gmail.com, k_praveen@cb.amrita.edu, gowrisanthya1805@gmail.com kssridhar697@gmail.com*

Abstract: In most of the organization, database management is the key component in the case of information infrastructure. In place of database management in-house, the computing industry has moved on the latest trend of outsourcing the database, like database as a service (DAS). Database outsourcing is storing the database in the third-party storage like cloud storage. The advantage of outsourcing is that the cost of managing the database and providing the security to that database is efficiently low. But this outsourcing has also come up with the security issues related to database. In the database management, one of the necessary requirement is to provide security to the database by holding the information confidential. To provide confidentiality to the database, the information in the database has to be encrypted and also has to be stored in the encrypted form. But when the database is encrypted there exists the problem of processing queries. Bucketization is one of the privacy preserving approach for executing SQL queries on encrypted data in DAS model. By partitioning and encrypting the attributes into query-able tables (buckets), the requested records can be distinguished. This paper shows that the response time analysis for querying data on order preserved encrypted (OPE) buckets is better than compared to querying other encrypted database models.

Keywords: Order preserving encryption scheme, Database as a service, Bucketization, Response time analysis

1. INTRODUCTION

In day-to-day life, the amount of usage of data is been increasing by the business organization which also in turn increases the storage size, cost of managing and maintaining these data. To fix this problem, the organizations started outsourcing the database, means the data is stored in a third-party database like Amazon Relational Database Service (RDS) and they will perform all the data management services. By storing the data in the third-party database, the security issues will raise. The privacy level of the database is not guaranteed while the data is stored in the third-party database because third party can access the sensitive information of the users. To provide security to these data's, the information stored in the third-party database should be encrypted which will prevent from the unauthorized access of user's data and disclosing the sensitive information. The database administrators can access the database without any privacy breach, because only the sensitive information will be encrypted. To access the encrypted data, the user has to decrypt that data with appropriate keys. For retrieving the data from the database that is been encrypted, usual method of applying normal queries is not been applicable.

The main motivation of this paper is to reduce the difficulty level of the user while retrieving the encrypted data from the remote server. Bucketization approach [1] is the one which facilitates query execution on normal or encrypted tables with false positives. After getting the results, each client will find all the unwanted data which is said to be false positives. Also, if the encrypted relations preserve order [2, 3], the server can execute the queries without any false positives. In this paper, the response time analysis of querying data on OPE buckets is done. This hybrid model (Bucketizing an order preserved encrypted relation) will reduce the client side processing by retrieving results with no false positives. Encrypting databases and its key management are open issues in database security. The first idea of the database system with property of having sub keys that allow the encryption and decryption of fields within a record based on Chinese Remainder Theorem [5] was proposed in 1981. The following papers [6, 7, 8, 9, 10, 11] also discuss about open issues related to database encryption and its privacy issues.

According to the type of data (e.g., text, image, audio, video, etc.) as well as on the type of search queries, the search over the encrypted data may vary. Consider Alice who outsources her database consisting of the following two relations:

1) TEACHER (tid, tname, salary, addr, did)

2) DEPT (did, dname, dch)

TEACHER table has attributes like teacher id, teacher name, salary, address and the id of the department. DEPT table has department id, department name, and name of the department chair. In a DAS model, these tables are stored by the service provider. Let us assume that the service provider is untrusted, so all the instances of these relations are encrypted and then stored on the server. It is important to remark that relational data can be encrypted at various levels.

At the table level the entire table is encrypted and stored, then at the row level each row is encrypted and stored in a cell, and at the attribute level the data of particular attribute is encrypted and stored in a cell. For time being, we follow row level granularity. The selected level of granularity will depend on the scheme which used to support the search, and also on system performance. Here we assume that data is encrypted at the row or tuple level. That is, corresponding rows of each table are encrypted as a single unit. In this case, the relational representation consists of a set of encrypted records. The database is now well suited for searching the SQL queries. For instance, Alice may wish to pose the following query to evaluate total salary for employees who work for Bob. In SQL, this query is written in the following way:

```
SELECT SUM (T. salary) FROM TEACHER as T, DEPT as D
WHERE T.did = D.did AND D.dch = “Bob”;
```

Now, if suppose Alice chooses an evaluation method where the query requests the encrypted form of the TEACHER and DEPT tables from the server. The client will then decrypt the tables and execute the query.

The rest of the paper is organized as follows. Hybrid model for querying is explained in section II. Experimental results are presented in section III. Concluding remarks are given in section IV.

2. HYBRID MODEL[4]

Fig. 1 explains query model that integrates bucketization and OPE. The hybrid model works in two phases:

1. Initially the data is bucketized using QOB algorithm [1] which calculates partitions points and minimum cost of bucketization.
2. Apply OPE scheme on each bucket.

QOB algorithm is used to query the encrypted database and to reduce the number of false positives, by selecting the bucket partitions which yield minimum cost. The boundaries of these bucketized attribute is stored as metadata in the client side.

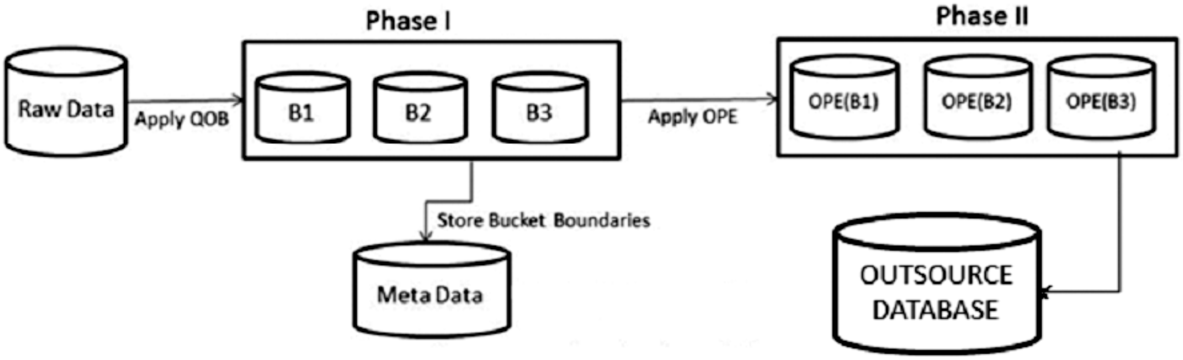
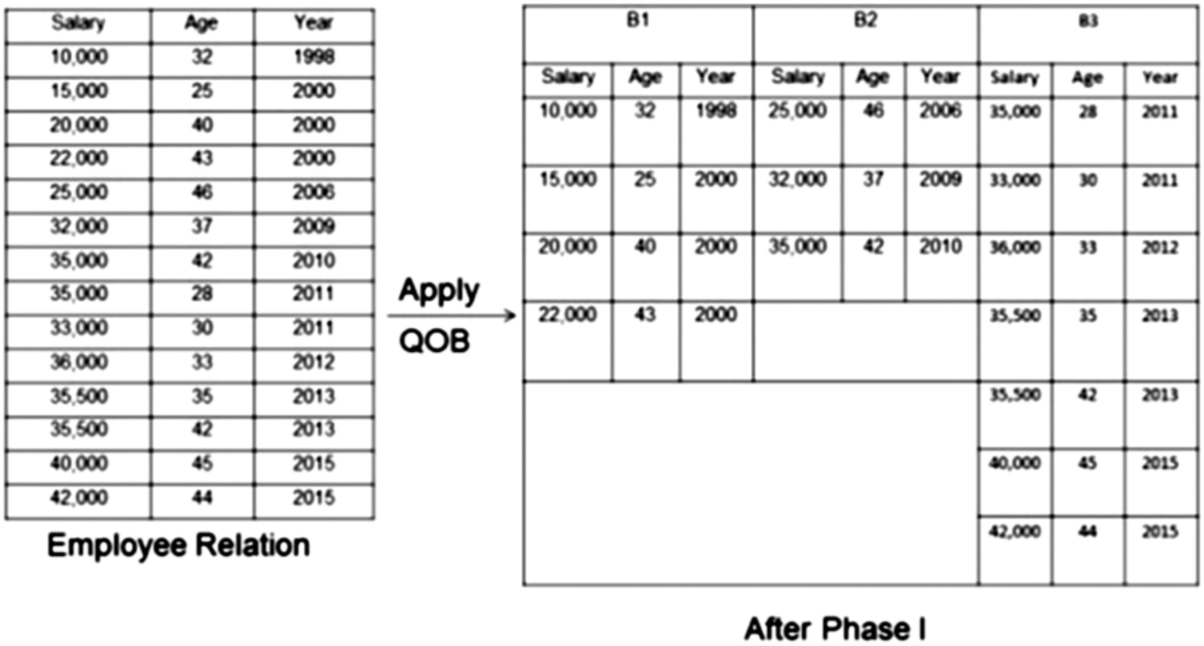


Figure 1: Hybrid model for querying

By considering the employee relation, we bucketized the data using attribute Year shown in Figure 2 with the partition size of 3. The input to the QOB algorithm is attribute value and frequency. In the discussed example at the end of Phase-I bucket B1 contains data corresponding to Year 1998 and 2000, B2 containing 2006, 2009 and 2010 and B3 containing 2011, 2012, 2013 and 2015. OPE protects the numerical order of the plaintext even after the encryption. A major disadvantage of encrypting the sensitive data by using the standard encryptions is that the data needs to be deciphered for query processing. Since OPE produces the cipher text that preserves the numerical ordering of the plaintexts [3]. A general OPE scheme with plaintext (resp. cipher text) space P (resp. C) is defined as, OPE = (RK, ENC, DEC) where RK is the random key generated using randomized key-generation algorithm, ENC and DEC are encryption and decryption process and DEC (ENC (RK, m)) = m. For every plaintext value m1, m2 in P and c1, c2 in C with key RK, the OPE property holds if m1< m2 then c1< c2. The adversary will not know about the plain text values without mapping of plain text and cipher text. This



Salary	Age	Year
10,000	32	1998
15,000	25	2000
20,000	40	2000
22,000	43	2000
25,000	46	2006
32,000	37	2009
35,000	42	2010
35,000	28	2011
33,000	30	2011
36,000	33	2012
35,500	35	2013
35,500	42	2013
40,000	45	2015
42,000	44	2015

Employee Relation

B1			B2			B3		
Salary	Age	Year	Salary	Age	Year	Salary	Age	Year
10,000	32	1998	25,000	46	2006	35,000	28	2011
15,000	25	2000	32,000	37	2009	33,000	30	2011
20,000	40	2000	35,000	42	2010	36,000	33	2012
22,000	43	2000				35,500	35	2013
						35,500	42	2013
						40,000	45	2015
						42,000	44	2015

After Phase I

Figure 2: Bucketized Relation

scheme supports both point and range queries (MIN, MAX, and COUNT). More number of queries can be directly processed without decrypting the data. Queries which include MIN, MAX will require only decryption of one record because the order is preserved. Few queries which include SUM, AVG requires the decryption of data. Consider a query which requests for details of employees joined between 1999 and 2001.

SELECT * FROM employee WHERE year > 1999 AND year < 2001;

This query is transformed into a query containing bucket identifier and the values (1999, 2001) are mapped into a cipher text with the corresponding bucket ids stored in metadata. So, the transformed query will be of the form:

SELECT * FROM employee WHERE Bid = B1 AND year > OPE (1999) AND year < OPE (2001);

With the help of this proposed hybrid querying model, the response time of query is reduced. The query results will also contain no false positives (exact match). Further updates or insertions to the table can be done without affecting the order of records which are already stored. Some of the queries do not require the decryption (COUNT).

3. EXPERIMENTAL RESULTS

The Amazon EC2 instance used for this experiment is a dedicated instance consists of a web application Encrypt DB used to query the encrypted outsourced databases. The application can perform standard encryption such as AES and Order Preserving Encryption. Python libraries are used for performing both encryptions like AES (pycrypto), OPE (pyope0.0.2) for comparison. PHP as server side scripting, is used to accept the requests from the client and the queries are sent to the outsourced database. Here we use MySQL Amazon RDS instance of class db.t2. micro as the outsourced database. The DB security group are defined to make the database accessible only the particular EC2 instance. The details of this instance are

Version: MySQL 5.6.27
Storage: 5GB
Endpoint: project-db.cuy3ai7fum7l.us-east-1.rds.amazonaws.com

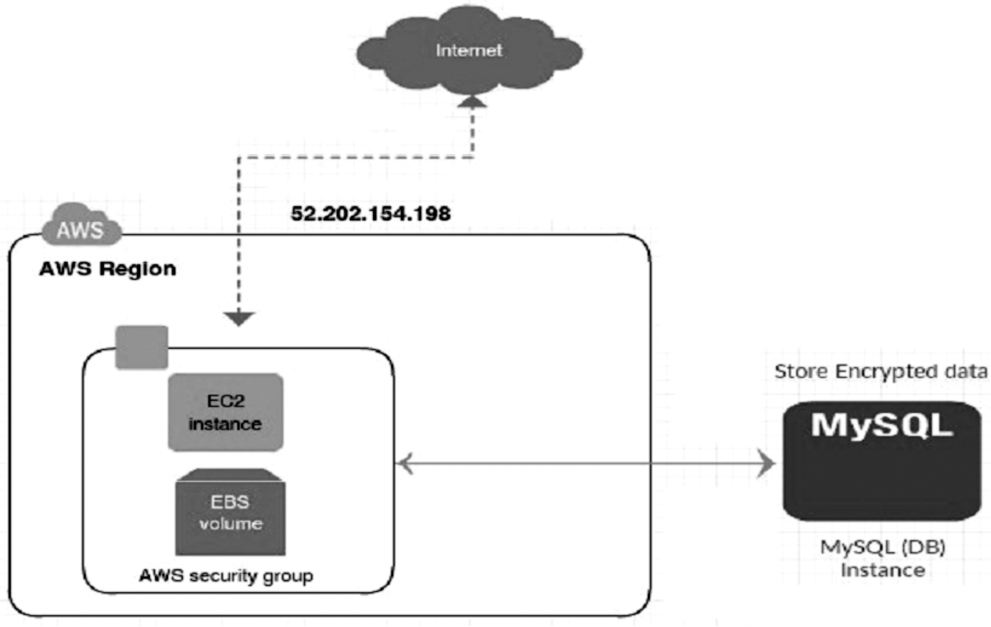


Figure 3: System Overview

For the experiment, the QOB algorithm was implemented and run for bucket sizes of M = 10 while results for OPE scheme over each size of M were simulated.

Three different experiments conducted:

1. AES + Bucketization: Data is encrypted with AES algorithm on the bucketized data.
2. OPE: Performance evaluation on order preserving encryption without bucketization.
3. OPE + Bucketization (Hybrid model): Applying OPE on each bucket.

While comparing with all these three experiments the querying using hybrid model give more performance as the result set always comprises exactly the tuples that were requested, i.e., no false positives. The response time for queries in hybrid approach is less compared to other schemes. For conducting the experiment, a data set consisting of 300,000 records of employee relation is selected and applied bucketization based on year attribute. A standalone Linux instance acts as the web server and is used to encrypting/ decrypting of data and query transformation is done using the metadata.

The metadata is stored in www directory of the instance. For every request the query will be transformed with the bucket id with the help of metadata and is sent to RDS instance where the encrypted data is stored. The query results are again sent back for decryption. Experiments were run on a t2. micro instance with 1 vCPU and 1 GB RAM with the clock speed 3.3 GHz. Figure 4 shows the response time taken for the queries on hybrid model and normal relation. The average time taken to execute each query is calculated in milliseconds (ms). The x-axis shows the bucketized attribute and the corresponding time taken on y-axis in ms. On encrypting the data with AES there exists a false positive which needs to be decrypted on client side to remove false positives which is not required in OPE encrypted databases. Figure 5 shows the number of AES encrypted records retrieved for different buckets. Retrieved results need to be decrypted at client side for removing false positives.

Figure 6 shows the decryption time in seconds for hybrid model is slightly higher for 2000 number of records when compared with OPE. In hybrid model, the server will not search from the first tuple instead it

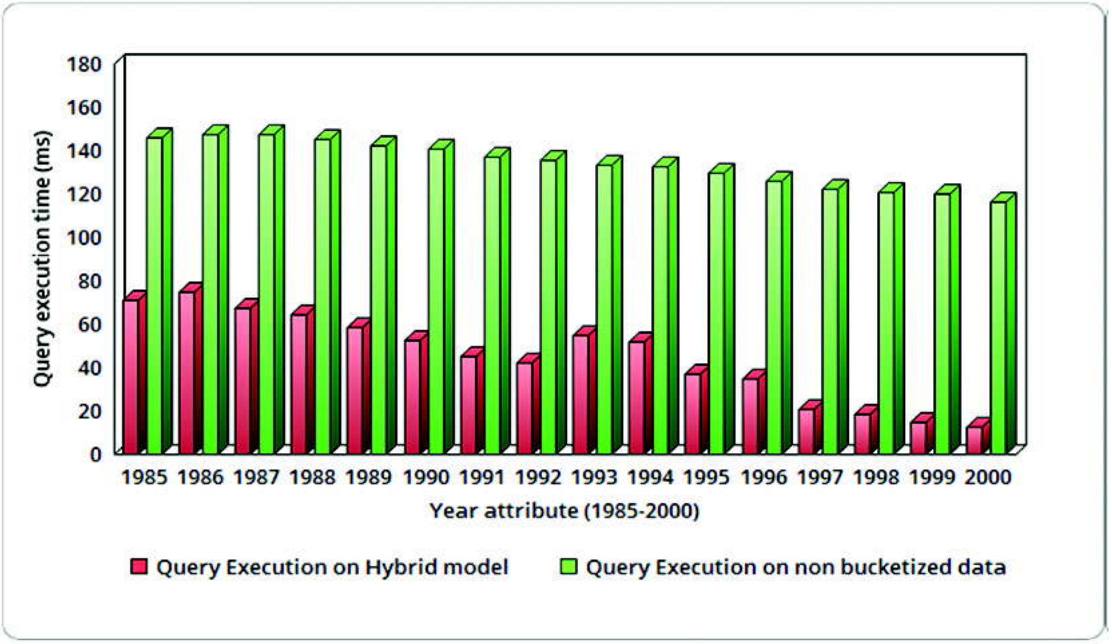


Figure 4: Query Encryption Time

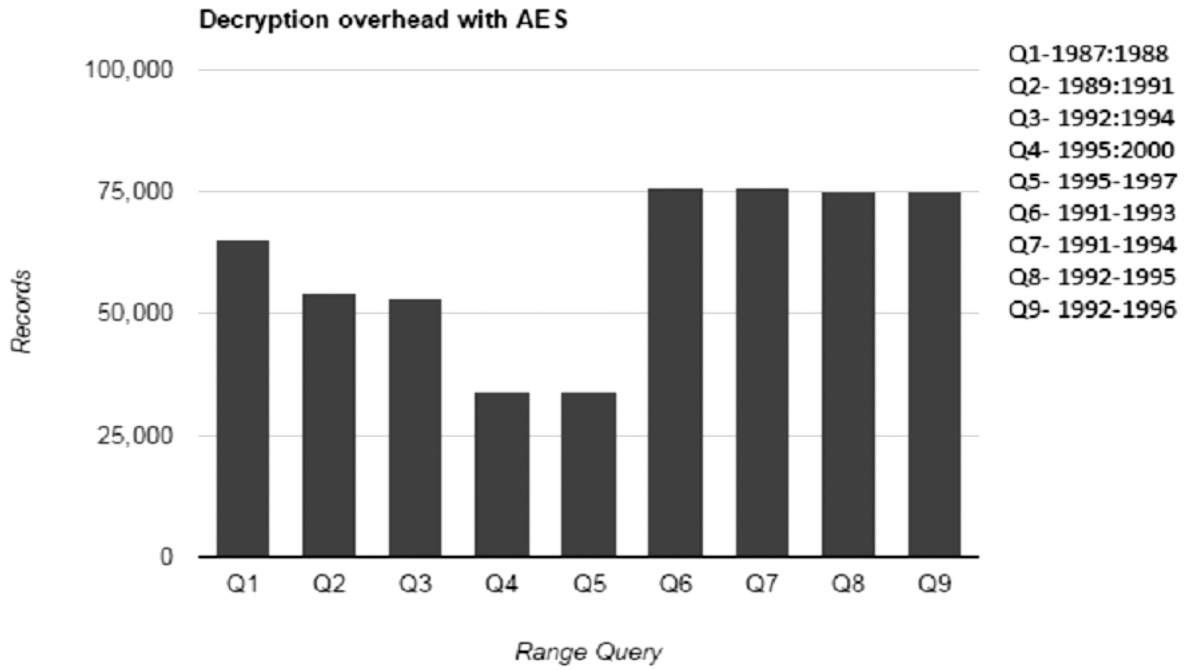
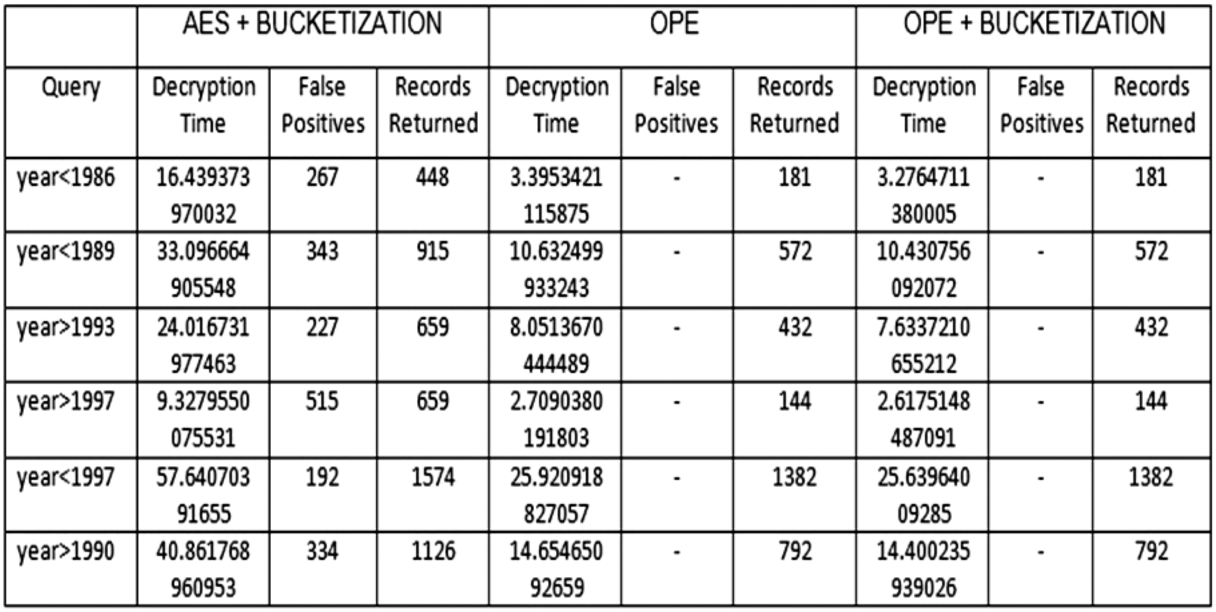


Figure 5: Records to be processed on client side



	AES + BUCKETIZATION			OPE			OPE + BUCKETIZATION		
Query	Decryption Time	False Positives	Records Returned	Decryption Time	False Positives	Records Returned	Decryption Time	False Positives	Records Returned
year<1986	16.439373970032	267	448	3.3953421115875	-	181	3.2764711380005	-	181
year<1989	33.096664905548	343	915	10.632499933243	-	572	10.430756092072	-	572
year>1993	24.016731977463	227	659	8.0513670444489	-	432	7.6337210655212	-	432
year>1997	9.3279550075531	515	659	2.7090380191803	-	144	2.6175148487091	-	144
year<1997	57.64070391655	192	1574	25.920918827057	-	1382	25.63964009285	-	1382
year>1990	40.861768960953	334	1126	14.65465092659	-	792	14.400235939026	-	792

Figure 6: Comparison of different querying schemes

looks into the exact bucket and retrieve the exact contents. In OPE the relation is not bucketized so the server need to search from the first tuple in an orderly manner to find the contents.

4. CONCLUSION

This paper shows that the combined usage of Bucketization and OPE technique improves the response time with no false positives when compared to the response time of independent queries on AES encrypted buckets and on OPE relations. One of the advantages of this hybrid model is, we can add a new tuple into the buckets without changing the encryption of other tuples. In future, we plan to design a database watermarking techniques [12] on buckets for ensuring the integrity of the OPE buckets. We also need to study on the issues such as encryption of non-numerical buckets and key management etc.

REFERENCES

[1] Bijit Hore, Sharad Mehrotra and Gene Tsudik. "A privacy-preserving index for range queries". In Proceedings of the Thirtieth international conference on Very Large Databases, VLDB Endowment, Vol. 30, pp 720-731, 2004.

[2] Boldyreva Alexandra, Nathan Chenette and Adam O'Neill. "Order Preserving Encryption revisited. Improved security analysis and alternative solutions". Advances in cryptology CRYPTO2011. Springer Berlin Heidelberg, pp 578-595, 2011.

[3] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant and Yirong Xu. "Order preserving encryption for numeric data". In Proceedings of the 2004 ACM SIGMOD international conference on Management of data. ACM, pp 563–574, 2004.

[4] Arjun K.R and Praveen. K. "A Hybrid Approach for Querying Numerical data in DAS-Model". IJCTA, No 8(5), pp 2077-2083, 2015

[5] Tzvi Chumash and Danfeng Yao. " Detection and prevention of insider threats in database driven web services". In Trust Management III, Springer, pp 117–132, 2009.

[6] George I Davida, David LWells and John B Kam. "A database encryption system with sub keys". ACM Transactions on Database Systems (TODS), Vol. 6(2), pp 312–328, 1981.

[7] Vahit Hakan Hacigumus. "Efficient key updates in encrypted database systems", Workshop on Secure Data Management. Springer Berlin Heidelberg, 2005.

[8] Dawn Xiaodong Song, David Wagner and Adrian Perrig. "Practical techniques for searches on encrypted data". In Security and Privacy, 2000. S&P 2000. Proceedings. IEEE, pp 44-55, 2000.

[9] Hakan H, Bala Iyer, Chen Li and Sharad Mehrotra. "Executing SQL over encrypted data in the database-service-provider model". In Proceedings of the 2002 ACM SIGMOD international conference on Management of data, ACM, pp 216-227, 2002.

[10] E Damiani, S De Capitani di Vimercati, S Paraboschi and P Samarati. "Computing range queries on obfuscated data". In Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), 2004.

[11] Einar Mykletun and Gene Tsudik. "Aggregation queries in the database-as-a service model". In Data and Applications Security XX, Springer, 2006, pp 89 –103.

[12] R. Arun, K. Praveen, D.C. Bose and H.V. Nath. "A distortion free relational database watermarking using patch work method". In Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012, pp. 531-538