A New Randomized Order Preserving Encryption Scheme

K. Srinivasa Reddy
Assistant Professor
Department of Computer Science and Engineering
University college of Engineering
Osmania University
Hyderabad, A.P., India.

S. Ramachandram, Ph.D.
Professor
Department of Computer Science and Engineering
University college of Engineering
Osmania University
Hyderabad, A.P., India.

ABSTRACT
Order Preserving Encryption (OPE) schemes have been examined to a great extent in the cryptography literature because of their prospective application to database design. OPE is an appealing method for database encryption as it permits to execute sort and range queries in an efficient manner without decrypting the data. Databases such as CryptDB are beginning to employ encryption to guard sensitive data. No existing OPE schemes that were proposed in the literature achieved IND-OCPA security except mutable Order-Preserving Encoding (mOPE) scheme, the first OPE scheme that satisfies IND-OCPA with respect to OPE encodings. However, mOPE scheme uses DET (deterministic encryption) to encrypt the plaintext values which leads to leakage of distribution of plaintext domain. This paper proposes a scheme called as Randomized Order Preserving Encryption (ROPE), a novel OPE scheme that leaks nothing beyond the order. ROPE follows the mOPE scheme by contributing randomness to it, so as to accomplish IND-OCPA security. The ROPE scheme implements insert, delete and query functions on an encrypted MySQL database. ROPE scheme permits various SQL queries to be employed instantly on encrypted data.The performance of ROPE scheme is compared with the existing DOPE scheme and observed that there is a query retrieval time overhead. Still, ROPE scheme renders more confidentiality and attains the IND-OCPA security for OPE when compared to the existing OPE schemes.

General Terms
Security, Algorithms

Keywords
ROPE, treeencoding, AVL tree, IND-OCPA, Trusted Proxy and RND

1. INTRODUCTION
Data stored in majority databases can be susceptible to attacks from vicious database administrators (DBAs). An opponent can hack into a database server to earn illegitimate access to sensitive data if the server is not secure. The database must be not only secure from the hackers; it also needs to be secure from the vicious DBAs access the data via an undocumented way. Furthermore, attackers can probably earn personal access to database servers and access data on disk.

One technique for shielding sensitive data is, to encrypt data stored in databases. Still, encryption schemes must be selected cautiously in order to permit database operations. Two popular operations are range queries and comparison queries. To protect information but permit such queries, an encryption scheme must be applied that holds order of values, but reveals nothing else.

Order-preserving encryption schemes help to give ciphertexts that maintain the numerical order of the plaintexts. Such schemes allow for storing encrypted data within a database, and the power to support popular operations such as sorting, range queries, and ranking. The perfect security objective for an OPE scheme, IND-OCPA (Indistinguishability under an Ordered Chosen-Plaintext Attack) in [8] is for the scheme to disclose nothing except for the order of the plaintexts. Many such schemes were developed by the researchers with different levels of success in the literature. Frequently, these schemes endangered security to maintain the order, revealing plaintexts of the data.

In [16] an OPE scheme called mutable order-preserving encoding (mOPE) was proposed that fulfills IND-OCPA with respect to encodings. The mOPE scheme functions by making a balanced search tree comprising all the encrypted data on a server, affording the server the power to order the tree based on the plaintexts without knowing any information about the plaintexts themselves. In the balanced search tree, all encrypted values are symbolized by a node. Each node in the tree has a binary encoding which symbolizes its path from the root to itself, permitting the server the power to preserve the order of the plaintexts. One disadvantage of mOPE is that, it uses DET for the ciphertexts. This means that the scheme can leak information about the distribution of plaintext values.

This paper introduces a novel scheme called Randomized Order Preserving Encryption (ROPE) which follows the mOPE scheme proposed in [16]. In ROPE scheme, DET is replaced with RND (random encryption) for the ciphertexts which prevents the leakage of information about the distribution of plaintext values. Since, the new scheme is based on RND for the ciphertexts, i.e. ciphertexts are randomized but it still maintains the plaintext order. Thus, the scheme is named as Randomized Order Preserving Encryption (ROPE) scheme. ROPE accomplishes the perfect security objective for OPE scheme called IND-OCPA, which demands that an adversary discovers nothing other than the order of plaintext data. It is showed that randomness is in reality needed for achieving IND-OCPA security. To show how ROPE can be employed in an application, it is implemented on an encrypted MySQL database to perform insert, delete and query operations.

The paper is organized as follows: Section 2 reports related work. Section 3introduces the security model. Section 4 elaborates ROPE scheme, Section 5assesses the performance of ROPE scheme. Section 6 concludes the paper.

2. RELATED WORK

Order-preserving symmetric encryption is a naive proposed in [1] for permitting effective range queries upon encrypted data. The first conventional cryptographic discussion of OPE appeared in the recent past in [8], where they formulated a security necessity for OPE and suggested a scheme that meets their security definition in an obvious and provable manner. In [8], they include that OPE schemes cannot meet the standard notion of security called indistinguishability against chosen-plaintext attack (IND-CPA), as OPE scheme is not only deterministic, but also leak the order-relations among the plaintexts.

The security option introduced in [8] is that of random order-preserving function (ROPF). ROPF requires that an OPE scheme appear "as-random-as-possible", although preserving the order. Nevertheless, OPE schemes that are classified under the ROPF notion proved to reveal at least half of the plaintext bits, besides order. Other OPE schemes either render less effective security definitions by giving premises about various attacks, which are impractical, or do not provide a security definition in any respect [2][3][4][5][7][9][10][11][12][13][14].

For the first time, in [16] a scheme called mutable order-preserving encoding (mOPE) is proposed, to attain perfect IND-OCPA security. In [16], it is shown that IND-OCPA in reality is accomplishable with mutable ciphertexts with respect to encodings. In [15] a new scheme called DOPE is proposed, which adopts mOPE scheme with few changes in the security model. In [15], the performance of DOPE scheme is compared with querying on plain text database and observed that there is a time overhead. CryptDB in [6] includes OPE in its design. CryptDB defends the database against attacks by running SQL queries over encrypted data.

3. SECURITY MODEL

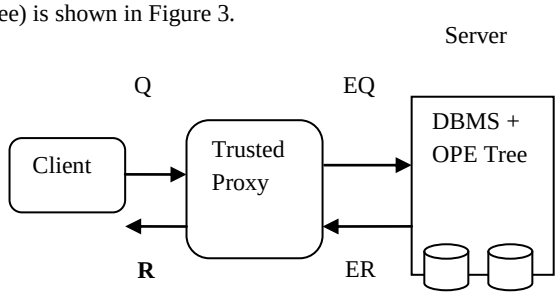
The security model depicted in Figure 1 shows the communication between the client, trusted proxy and server. The client is the owner of the data to be encrypted, a trusted proxy which encrypts the client's plain data and decrypts the encrypted result sent by the server. The server is considered as a passive adversary. The server follows the ROPE algorithm honestly and gives exact results to the client via a trusted proxy. The server simply tries to understand the information about the data besides order.

4. RANDOMIZED ORDER PRESERVING ENCRYPTION (ROPE) SCHEME

Take into account a ROPE scheme that will attain the goal of precise security. Imagine a client wishes to encrypt the following elements: 56, 36, 71, 26, and 41. The resultant order of OPE encodings for the individual elements is 4, 2, 6, 1, and 3. The encodings are accurate as it conveys the server simply the order of the elements.

ROPE functions by constructing a balanced search tree called an AVL tree. An AVLsearch tree consists of a root node k, the nodes lesser than k are placed in the left part of the tree and the nodes greater than k are placed in the right part of the tree. Each node in the AVL search tree as depicted in Figure 2 holds the RND value, and cipher texts are inserted into the tree in the order of the plaintext values through a trusted proxy.

Consider the setting in which the client inserts 66 into the server. At the beginning the proxy demands the AVL tree for the root node, and the AVL tree returns RND (56). The trusted proxy decrypts RND (56) and compares 56 with 66, since 66 greater than 56 the proxy demands the AVL tree for the right child of the root node. The AVL tree responds with RND (71). The proxy decrypts RND (71) and compares 71 with 66, since 66 lesser than 71 then proxy demands the AVL tree for the left child of the new node. The AVL tree returns null. Now the proxy will insert RND (66) at that position. The interaction between the trusted proxy and the AVL tree (OPE tree) is shown in Figure 3.



Q- Query, R- Result,

EQ- Encrypted query, ER- encrypted result.

Fig 1: Security model

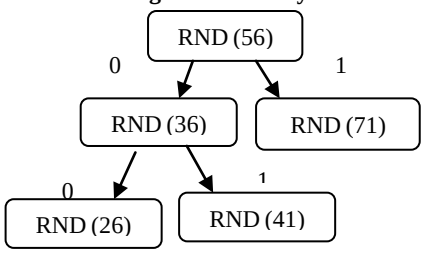


Fig 2: AVL tree (OPE tree)

The paths of the tree are referred using either 0 or 1, where a '0' refers the left edge and '1' refers the right edge respectively. The encoding of each plain text value is the path from the root node to that value. The root node (56) path is an empty string. Add all paths from root to each node of the tree to the same length by using the rule: tree encoding = [path length] 10. . .0. The tree encodings for each plaintext value is depicted in Figure 4.

The length of tree encodings is directly proportional to the height of the AVL tree. So, there is a need for AVL Tree balancing which leads to changes in the tree encodings. The OPE tree server modifies each of the tree encodings in the table every time that a tree balancing takes place.

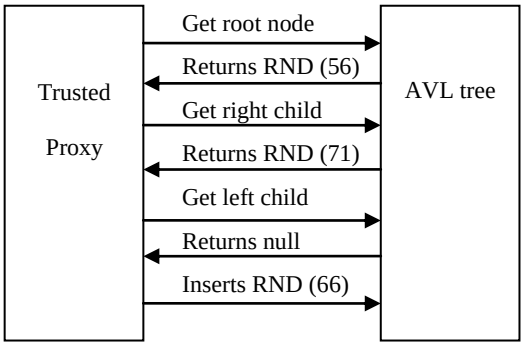


Fig 3: Interaction between trusted proxy and AVL tree

Ciphertext	Plaintexts	Tree encoding
	[Path length] 10...0	
RND (56)	56	[]100 = decimal 4
RND (36)	36	[0]10 = decimal 2
RND (71)	71	[1]10 = decimal 6
RND (26)	26	[00]1 = decimal 1
RND (41)	41	[01]1 = decimal 3

Fig 4: Treeencodings for the corresponding plain text values

A Randomized Order-Preserving Encryption scheme is defined as ROPE= (K, Enc, Dec) executed by a trusted proxy P and a server S, where K is the key, Enc is encryption and Dec is decryption. Enc and Dec are interactive.

ROPE algorithm works as follows:

1. K is generated randomly at P : K=RND.key
2. Enc(V, K) runs at P and S:

- P calculates C←Enc (V, K) and sends C to S.
- If C is present in S, S returns encoding E of C to P
- Else go to step 3.

3. AVL Tree searches for V as follows:

- P asks for the root of AVL tree to get ciphertext C′. P obtains V′ by decrypting C'.
- P orders AVL tree, if V <V′ go towards "left" else go towards "right".
- Repeat until AVL tree finds a node with a null value in the tree.
- After inserting C into the AVL tree, it gets the tree encoding of C and then calculates E of C and puts it in database table.
- Tree server automatically updates the corresponding tree encodings in the table whenever a tree balancing occurs.

4. Dec(E, K) runs at S and P:

- S returns E to P as a result of query retrieval.
- P communicates with S to get ciphertext C for the corresponding E
- P returns v←Dec(C, K) to the client.

4.1 Correctness of ROPE

The ROPE scheme outputs the correct order of ciphertexts with respect to plaintexts and produces correct decrypted values.Consider a plain text domain P_D, the ROPE scheme is said to be correct for P_D if for all secret keys K.

- $\forall V$: $V \in P_D$ and for each ciphertext C, Dec(C, K) = V; and,
- $\forall S$: $S= \{V_1,\ldots\ldots,V_N\}$, $S \in P_D$, for any plain text pair of S, for all ciphertexts pair received, the result is if $V_I < V_J$ then $C_I < C_J$.

4.2 Security of ROPE

The exact security goal of ROPE is that it should reveal nothing in addition to order. The security definition of ROPE is the IND-OCPA definition introduced in [8]. The definition states that an adversary cannot make out between pairs of ciphertext values provided that the pairs have the same order relation.

The justification for why ROPE scheme is IND-OCPAsecure is that:The mOPE [16] scheme uses DET for the ciphertexts. Since DET is a deterministic scheme, where every encryption of a plaintext value V would be mapped to the same ciphertext C. Once the adversary decrypts C, all encryptions of V are revealed. Hence, the ciphertexts in ROPE scheme employs RND. In RND, each randomized OPE encryption of V maps to a different Cr, and decryption of a specific Cr does not give total assurance in decrypting all encryptions of V. So, now every time we request a value V as part of insert, delete, or query, the proxy encrypts it with new randomness using RND, thus accomplishingIND-OCPA security goal of ROPE scheme.

5. PERFORMANCE OF ROPE

This section discusses the performance of ROPE on an encrypted database. ROPE is capable of allowing order operations effectively on an encrypted database without modifying the database server software.

ROPE functions on a database by considering the following settings:

- The trusted proxy executes ROPE algorithm to get encoding values.
- The server implements separate functions defined by ROPE algorithm such as insert and delete.
- For any insertion operation, a ciphertext Cfor V is already in the AVL Tree, the tree server simply increments node's counter value which is initially set to zero. Otherwise, the server will inserts C into the AVL tree with counter set to 1 and stores it in table.
- During the deletion process, the server decrements node's counter value. If the counter value becomes zero, the node is deleted from the tree and also removes it from the table.

ROPE algorithm is implemented in Java over MySQL database server. The experiments were conducted using Windows 7 with a 2.27 GHz Intel Core i5 processor and 2GB of memory.AES in CBC mode along with a random initialization vector (IV) is used to get RND values.

ROPE scheme performance is examined by considering various SQL queries which are run on a set of records from 100 to 1000 by incrementing 100 records each time. ROPE scheme supports different SQL queries such as range queries, equality queries, join queries and aggregate functions (count, min and max).

Table 1. applying the query: "select * from employee where salary>70000" on employee table by incrementing the records by 100

Records	Retrieval Time in ms			No. of records retrieved
	DOPE	ROPE	Plain data	
100	1913	1957	15	28
200	4708	4847	15	70

300	5819	5983	16	110
400	9003	10064	16	150
500	11138	12244	16	183
600	11310	12406	16	209
700	15615	16025	16	248
800	16614	18126	15	285
900	18970	22845	15	325
1000	22074	25257	16	370

During the experiment, a range query is run on a set of 100, 200, 300….1000 records present in an encrypted database and the corresponding retrieval time which includes the communication time taken between the proxy and the tree is calculated. The results are described in Table 1. Here, the total retrieval time taken is the sum of communication time and the retrieval time on encrypted data.

The above mentioned range query is considered again on 1000 records by changing the range query condition (i.e. N) and the results are depicted in Table 2. As shown in Table 2, for existing records such 40000, 50080, 80061 and 90122 in the table, the communication time is zero because the encodings of these values are directly acquired from the table. The other values those are not present in the database table takes additional time to get encodings of N. The time variation between retrieval times in milliseconds excluding communication time taken on encrypted data and unencrypted data are clearly depicted in Table 3 and Table 4. The comparisons between the performance of ROPE algorithm and DOPE [15] algorithm are shown in Figure 5, Figure 6, Figure 7, Figure 8and Figure 9. From the results, it is observed that ROPE is taking more query retrieval time when compared to DOPE. Still, ROPE is more secure than DOPE and may trade the time overhead for increased security.

Table 2. Applying the query: "select * from employee where salary>N" on employee table with 1000 records

N	Communication time in ms		Total Retrieval Time in ms		No. of records retrieved
	DOPE	ROPE	DOPE	ROPE	
10000	23634	26410	24430	27213	1000
20000	21169	26725	21918	27528	1000
30000	23431	26044	24148	26876	1000
40000	0	0	671	683	988
50080	0	0	515	525	844
60000	22698	25847	23010	26270	617
70000	21887	25973	22074	26257	370
80061	0	0	93	181	185
90000	22885	26230	22948	26407	84
90122	0	0	47	142	83

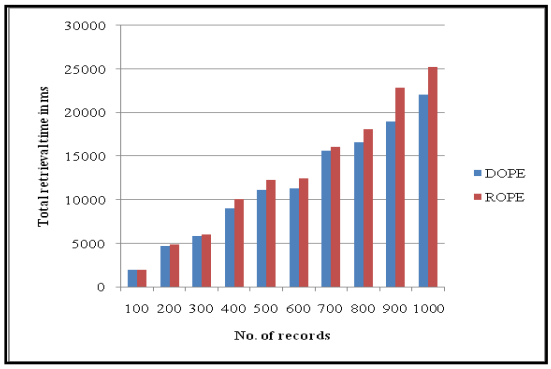


Fig 5: select * from employee where salary>70000

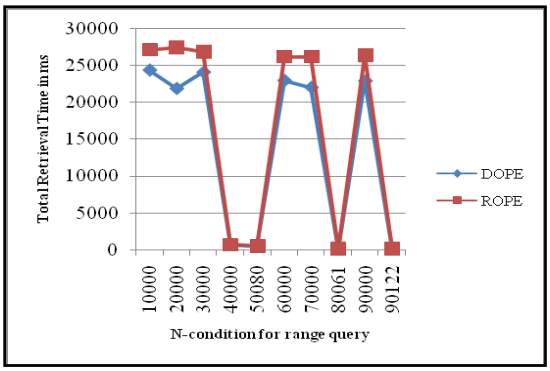


Fig 6: select * from employee where salary>N by taking total retrieval time

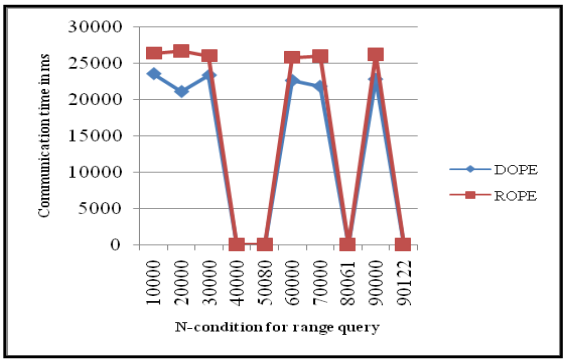


Fig 7: select * from employee where salary>N by taking communication time

Table 3. Retrieval Time in ms excluding communication time for the query: "select * from employee where salary>N" on 1000 records

N	Retrieval Time in ms excluding communication time			No. of records retrieved
	DOPE	ROPE	Plain data	
10000	796	803	15	1000
20000	749	803	15	1000
30000	717	832	16	1000
40000	671	683	15	988
50080	515	525	16	844
60000	312	423	16	617
70000	187	284	16	370
80061	93	181	16	185
90000	63	177	15	84
90122	47	142	15	83

Table 4. Retrieval Time in ms excluding communication time for the query: "select * from employee where salary>70000"

Records	Retrieval Time in ms excluding communication time			No. of records retrieved
	DOPE	ROPE	Plain data	
100	43	70	15	28
200	57	74	15	70
300	63	90	16	110
400	95	107	16	150
500	109	115	16	183

600	125	154	16	209
700	140	202	16	248
800	156	231	15	285
900	157	262	15	325
1000	187	284	16	370

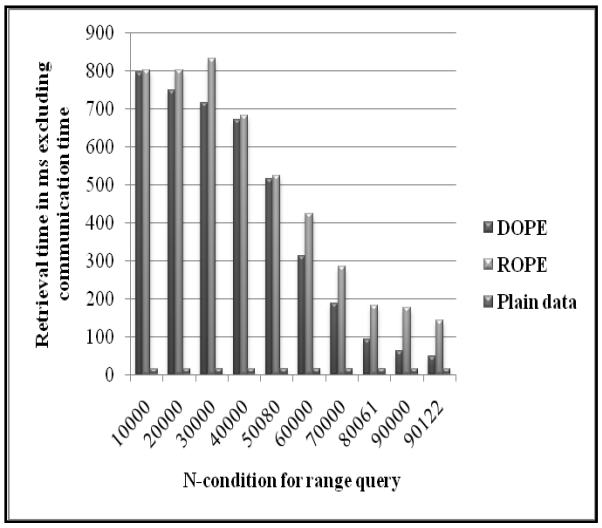


Fig 8: Retrieval Time in ms excluding communication time for the query: "select * from employee where salary>N" on 1000 records

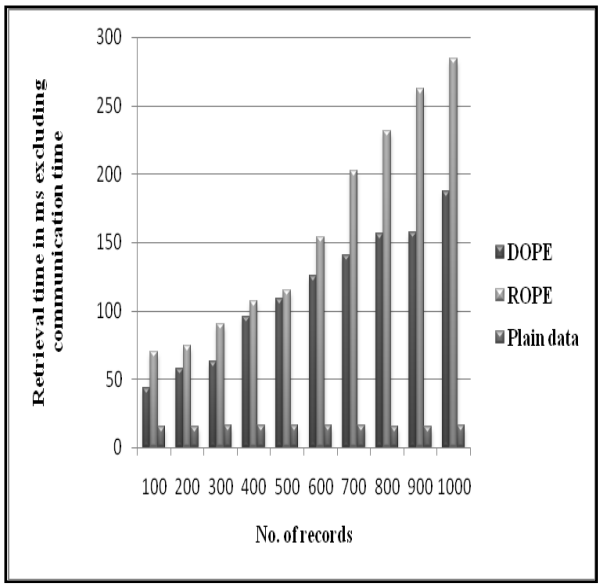


Fig 9: Retrieval Time in ms excluding communication time for the query: "select * from employee where salary>70000"

6. CONCLUSION

In the ROPE scheme, an adversary learns nothing beyond the order of plaintext data. ROPE follows the mOPE scheme by contributing randomness to it, so as to accomplish IND-OCPA security. The ROPE scheme is based on RND for the ciphertexts, where RND is probabilistic, implies that two equal values are mapped to different ciphertexts with extreme probability. It allows efficient and secure queries to be right away employed on encrypted data. ROPE scheme can be employed on various types of queries such as range queries, aggregate functions (count, min and max), equality queries, and join queries.

The performance of ROPE scheme is compared with the existing DOPE scheme and observed that there is a query retrieval time overhead.Still, ROPE scheme renders more confidentiality and attains the IND-OCPA security for OPE when compared to the existing OPE schemes. In the future, we will reduce the query retrieval time caused due to the communication between the proxy and tree server and also further examine the security issues of OPE.

7. REFERENCES

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," In Proceedings of the ACM SIGMOD international conference on Management of data, pp. 563-574, 2004.

[2] H. Kadhem, T. Amagasa, and H. Kitagawa, "A secure and efficient order preserving encryption scheme for relational databases," International Conference on Knowledge Management and Information Sharing, Spain, October 2010.

[3] S. Lee, T. J. Park, D. Lee, T. Nam, and S. Kim, "Chaotic order preserving encryption for efficient and secure queries on databases," IEICE transactions on information and systems, vol. 92(11), pp. 2207-2217, 2009.

[4] D. Liu and S. Wang, "Programmable order-preserving secure index for encrypted database query," IEEE 5th International Conference on Cloud Computing, pp. 502-509, 2012.

[5] D. Liu and S. Wang, "Nonlinear order preserving index for encrypted database query in service cloud environments," Concurrency and Computation: Practice and Experience, Wiley Online Library, vol. 25(13), pp. 1967-1984, 2013.

[6] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting confidentiality with encrypted query processing," In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, pp. 85-100, 2011.

[7] L. Xiao, I.L. Yen, and D. T. Huynh, "Extending order preserving encryption for multi-user systems," IACR Cryptology ePrint Archive, Report 2012/192, 2012.

[8] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order-preserving symmetric encryption," Advances in Cryptology-EUROCRYPT, pp. 224-241, 2009.

[9] D. Agrawal, A. El Abbadi, F. Emekci, and A. Metwally, "Database management as a service: Challenges and opportunities," In IEEE 25th International Conference on Data Engineering, ICDE, pp. 1709-1716, 2009.

[10] G. W. Ang, J. H. Woelfel, and T. P. Woloszyn, "System and method of sort-order preserving tokenization," US Patent Application 13/450, 2012.

[11] H. Kadhem, T. Amagasa, and H. Kitagawa, "MV-OPES: Multivalued-order preserving encryption scheme: A novel scheme for encrypting integer value to many different values," IEICE Trans. on Info. And Systems, vol. 93(9), pp. 2520-2533, 2010.

[12] G. Ozsoyoglu, D. A. Singer, and S. S. Chung, "Anti-tamper databases: Querying encrypted databases," In DBSec, pp. 133-146, 2003.

[13] L. Xiao, I.L. Yen, and D. T. Huynh, "A note for the ideal orderpreserving encryption n object and generalized order-preserving encryption," IACR Cryptology ePrint Archive, Report 2012/350, 2012.

[14] D. Yum, D. Kim, J. Kim, P. Lee, and S. Hong, "Order-preserving encryption for non-uniformly distributed plaintexts," In Intl. Workshop on Information Security Applications, pp. 84-97, 2012.

[15] K.Srinivasa Reddy, Sirandas Ramachandram, "A Novel Dynamic Order-Preserving Encryption Scheme," in press, IEEE First International Conference on Networks & Soft Computing, August, 2014

[16] R.A. Popa, F. Li, N. Zeldovich, "An Ideal-Security Protocol for Order Preserving Encoding," IEEE Symposium on Security and Privacy (SP), pp. 463-477, 2013.

[17]