International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 5 2383 - 2388

Query Processing on Encrypted Database: A Performance Evaluation

Rajendra H.Rathod
PG Student, Computer Science & Engg. Deptt
Prof.Ram Meghe Institute of Technology & Research,
Badnera-Amravati, Maharashtra, India
rh_rathod@yahoo.com

Dr.C.A.Dhote
Professor, Computer Science & Engg. Deptt
Prof.Ram Meghe Institute of Technology & Research,
Badnera-Amravati, Maharashtra, India
vikasdhote@rediffmail.com

Abstract **-** Due to the rapid development of the global internetworked infrastructure, the research scope of secure data management has been greatly expanded. Now due to the openness of the internet, databases are accessed by legitimate authorized users as well as the outsiders. So the problems in traditional database security need to be revisited and readdressed in decentralized web-based and open environments. Secure and efficient algorithms are needed that provide the ability to query over encrypted database and allow optimized processing of data. Clearly, there is a compromise between the degree of security provided by encryption and the efficient querying of the database, because the operations of encryption and decryption greatly degrade query performance. We evaluate the performance of the query processing over encrypted database with algorithms REA and with most common algorithms: AES and RC6. The performance of the query over encrypted databases using AES, RC6 and REA algorithms are compared. This performance measure was conducted in terms of query execution time. The experiment results show the advantages of the algorithm REA over other algorithm AES and RC6 in terms of the query execution time. The results show that the encryption algorithm REA outperforms other encryption algorithms at performance and security in databases. The performance of the query enhanced if we encrypt the database by using REA algorithm. So it has achieved security requirements and is fast enough for most applications. REA algorithm is limiting the added time cost for encryption and decryption to do not degrade the performance of a database system.

Keywords: *Encryption, Retrieval, Query, AES, REA, RC6*

*****

I. INTRODUCTION

We will find two types of threats to data security. The threat is from the outsiders, unauthorized and unauthenticated users. There is a difference between unauthenticated and unauthorized users. Authentication means that a user has got license to access a particular data. Authorization comes after authentication. Authorization means that after being authenticated, what are the permissions allowed to that user, up to what extent he has got the authority, whether he can only read that data or can make changes into it. Although outsider threats are very harmful, it is the insider threats which take place more. An insider threat includes insider employees who leak sensitive information, using their authority to harm the sensitive data for example, in a database system, financial data is often considered confidential, and hence only authorized persons are allowed to access such data. So we put such an encryption algorithm which is reliable and efficient also. We are focusing on the evaluation of query on encrypted database. This paper observes a method for evaluating query processing performance over encrypted database with encryption algorithms like AES, RC6 and REA

Solution to secure the unsecured data: Encryption

To secure the unsecured data one can provide the solution called encryption. Encryption is one of the solutions to provide security to the data from database. But no mechanisms are strong and reliable enough to protect data from threats to security. So encrypt data in a database we design a encryption algorithm which protect the data from other unauthenticated users. Cryptographic algorithms are used to encode a message from its unencrypted state into an encrypted message. The three primary methods are the hashing, symmetric, and asymmetric methods. Symmetric algorithms require both ends of an encrypted message to have the same key and processing algorithms. Symmetric algorithms generate a secret key that must be protected. A private key is simply a key that is not disclosed to people who are not authorized to use the encryption system. While asymmetric algorithms use two keys to encrypt and decrypt data. These keys are referred to as the public key and the private key. The public key can be used by the sender to encrypt a message, and the private key can be used by the receiver to decrypt the message.

In this paper we put forward the encryption algorithm REA. It is efficient and secure. It has accomplished security and it is fast enough for most widely used software. The proposed algorithm REA limits the added time cost for encryption and decryption and at the same time does not degrade the performance of a database system. [1], [3]. This paper observes a method for evaluating the performance of query processing on encryption algorithms AES, RC6 and REA. Then we will compare the output of time elapsed using these algorithm to process the query, and evaluate the performance. RC6, which is a simple, fast, and secure block cipher, was the final candidate algorithm in the AES project of the United States and the NESSIE project of Europe. These projects require 128-bit and variable-length block cipher encryption algorithm. RC6 has a modified Feistel structure and a disadvantage that it has different algorithm between encryption and decryption. Thus, the RC6 algorithm needs double space compared with the same structure of encryption and decryption when it is implemented on hardware. [9]

II. ENCRYPTION ALGORITHMS

RC6: We recommend the encryption algorithm "Rivest Cipher 6" algorithm RC6 to encrypt the database. RC6 is one of the best AES algorithm and simple cipher used for data security. RC6 is a very simple cipher with excellent security credentials. RC6 has a block size of 128 bits and supports key sizes of 128,

 ISSN: 2321-8169
2383 - 2388

192 and 256 bits and 20 rounds. It was designed in order to meet the requirements of the AES. It is an improvement of the RC5 algorithm. It makes use of 4 registers A,B,C,D each of 32 bit which contains the initial input plaintext as well as the output cipher text at the end of encryption. RC6 algorithm has a modified Feistel structure and presented symbolically as RC6-w/r/b. w means 32-bits as the size of word, r denotes the number of round. The first byte plain text of cipher text is placed in the least significant byte of A, the last byte of plain text or cipher text is placed into the most significant byte of D [11],[17],[15]. The operation used in RC6 are defined as follows: [3],[4],[6],[11]

- A+B integer addition modulo 2w
- A-B integer subtraction modulo 2w
- A⊕B bitwise exclusive-or of w-bit words
- A B integer multiplication modulo 2w
- A⋘B rotation of the w-bit word A to the left by the amount given by the least significant log w bits of B
- A⋙B rotation of the w-bit word A to the right by the amount given by the least significant log w bits of B
- (A,B,C,D)=(B,C,D,A) parallel assignment of values on the right to registers on the left. [17]

Encryption algorithm of RC6-w/r/b

Input: Plain text stored in four w-bit input registers
A, B, C, D
Number of r rounds
w-bit round keys S[0,…,2r + 3]
Output: Cipher text stored in A, B, C, D
Procedure: B = B + S[0]
D = D + S[1]
for (i=1; i<r; i++)
{
t = (B (2B + 1)) ⋘ log w
u = (D (2D + 1)) ⋘ log w
A = ((A ⊕ t) ⋘ u) + S[2i]
C = ((C ⊕ u) ⋘ t) + S[2i+1]
(A, B, C, D) = (B, C, D, A)
}
A = A + S[2r+2]
C = C + S[2r+3]

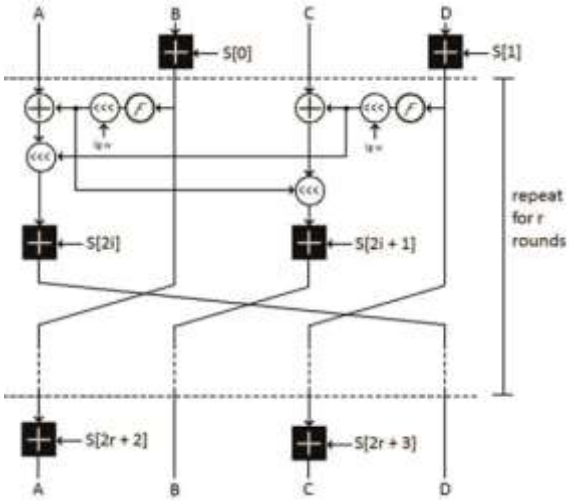

Figure1: RC6 algorithm

Decryption algorithm of RC6-w/r/b

Input: Plain text stored in four w-bit input registers
A, B, C, D
Number of r rounds
w-bit round keys S[0,…,2r + 3]
Output: Cipher text stored in A, B, C, D
Procedure: C = C - S[2r+3]
A = A + S[2r+2]
for i=1 downto 1 do
{
(A, B, C, D) = (D, A, B, C)
u = (D * (2D + 1)) <<< log w
t = (B * (2B + 1)) <<< log w
C = ((C - S[2i + 1]) >>> t) ⊕ u
A = ((A - S[2i]) >>> u) ⊕ t
}
D = D - S[1]
B = B - S[0]

Reverse Encryption Algorithm

We recommend the encryption algorithm which encrypts the data of database called REA. REA is limiting the added time cost that is delayed time for encryption and decryption not degrading the performance of a database system [3],[4],[6]. REA takes a variable length key, making it ideal for securing data. The REA algorithm works like symmetric algorithms. We will add the keys to the text while encryption and remove the keys from the text in the decryption. The details and working of the proposed algorithm REA are given below [3], [4], [5], [32]

Encryption steps

1: Input and add the text and the key.
2: Convert this text to ascii code.
3: Convert this ascii code to binary data.
4: Reverse this binary data that is 1 to 0 and 0 to 1.
5: Take each 8 bits from this binary data and obtain the ascii code from it.
6: Divide this ascii code by 4 and put it as one character
7: Obtain the remainder of the previous divide and put it as a second character.
8: Return encrypted text.

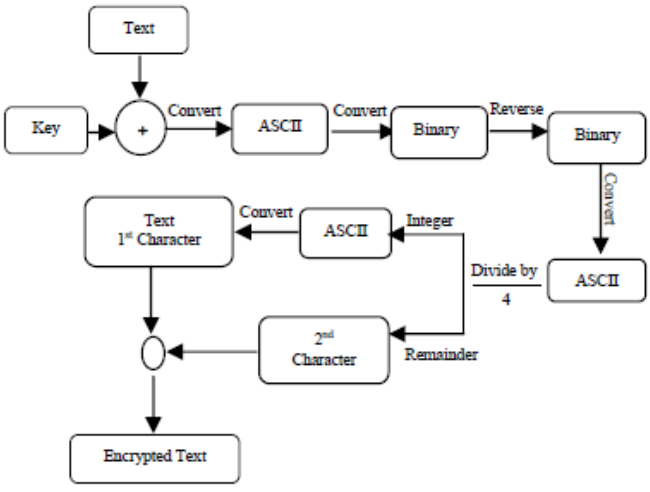

Figure 2: REA encryption algorithm

Encryption Algorithm

INPUT: Plaintext (StrValue), Key (StrKey).
OUTPUT: Ciphertext (EncryptedData).

1: Add the key to Text (StrKey + StrValue)---> full string (StrFullVlaue).
2: Convert the Previous Text(StrFullVlaue) to ascii code (hexdata).
3: Foreach (byte b in hexdata).
a. Convert the Previous ascii code (hexdata) to binary data (StrChar).
b. Switch (StrChar.Length).
Case 7 → StrChar = "0" + StrChar.
Case 6 → StrChar = "00" + StrChar.
Case 5 → StrChar = "000" + StrChar.
Case 4 → StrChar = "0000" + StrChar.
Case 3 → StrChar = "00000" + StrChar.
Case 2 → StrChar = "000000" + StrChar.
Case 1 → StrChar = "0000000" + StrChar.
Case 0 → StrChar = "00000000" + StrChar.
c. StrEncrypt += StrChar. (where, StrEncrypt= "")
4: Reverse the Previous Binary Data(StrEncrypt).
5: For i from 0 to StrValue.Length do the following:
a. if (binarybyte.Length == 8).
i.Convert the binary data (StrEncrypt) to ascii code and,
ii.Divide the ascii by 4 → the result(first character) and,
iii.The remainder of the previous → second character.
6: Return (EncryptedData). [3], [4],[5],[32]

Decryption steps

1: Input the encrypted text and the key.
2: Loop on the encrypted text to obtain ascii code of characters and add the next character.
3: Multiply ascii code of the first character by 4.
4: Add the next digit (remainder) to the result multiplying operation.
5: Convert this ascii code to binary data.
6: Reverse this binary data that is 1 to 0 and 0 to 1
7: Get each 8 bits from this binary data and obtain the ascii code from it.
8: Convert this ascii code to text.
9: Remove the key from the text.
10: Return decrypted data. [3], [4], [5], [32]

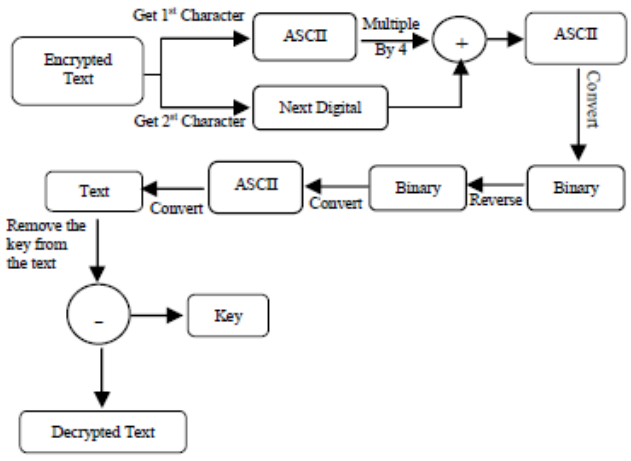

Figure 3: REA decryption algorithm

Decryption Algorithm

INPUT: Ciphertext (EncryptedData), the Key (StrKey).
OUTPUT: Plaintext (DecryptedData),
1: For (i = 0; i < EncryptedData.Length; i += 2)
a. Get the ascii code of the encrypted text
b. newascii = (EncryptedData[i] * 4) + the next digit(remainder)[i+1].
2: Foreach (byte b in newascii).
a. Convert the Previous ascii code (newascii) to binary data (StrChar).
b. Switch (StrChar.Length).
Case 7 → StrChar = "0" + StrChar.
Case 6 → StrChar = "00" + StrChar.
Case 5 → StrChar = "000" + StrChar.
Case 4 → StrChar = "0000" + StrChar.
Case 3 → StrChar = "00000" + StrChar.
Case 2 → StrChar = "000000" + StrChar.
Case 1 → StrChar = "0000000" + StrChar.
Case 0 → StrChar = "00000000" + StrChar.
c. StrDecrypt += StrChar.
3: Reverse the Previous Binary Data(StrDecrypt).
4: For i from 0 to StrDecrypt.Length do the following:
a. if (binarybyte.Length == 8).
i. Convert the binary data (StrChar) to ascii code (hexdata) and,
ii. Convert the previous ascii code (hexdata) to the text (StrFullVlaue).
5: Remove the key from the text (StrFullVlaue - StrKey) → (StrValue).
6: Return (DecryptedData). [3], [4], [5], [32]

III. System Design

Query Processing is costly means that it takes more time for encryption and decryption. Our aim is to provide maximum security to database at the same time give best performance. So we design a system that will take less time and provide security to the database. We design the database for query processing.

1. Create four databases
2. Keep the first database non-encrypted.
3. Encrypt second database with AES encryption algorithm.
4. Encrypt third database with RC6 encryption algorithm.
5. Encrypt fourth database with REA encryption algorithm.
6. Then we apply same queries on these four databases.

We examined query processing performance evaluation over encrypted databases with the proposed algorithm (REA). The performance measure of query processing is conducted in terms of query execution time.

IV. System Implementation

The system is implemented by available hardware and software resources. The software resource used in the system is the PHP, WAMP, which is an open source. In the experiments, we used four databases from the database "project" are:

1. tblenc_plain has not any encrypted fields.
2. tblenc_aes has encrypted some fields with AES encryption algorithm.
3. tblenc_rc6 has encrypted same fields with RC6 encryption algorithm.
4. tblenc_rea has encrypted same fields with REA encryption algorithm.

ISSN: 2321-8169
2383 - 2388

Now, we started executing the queries on these databases. Every query from the first to the fifth executes on the database "tblenc_plain" then calculates the execution time and repeats executes on the database "tblenc_aes" then calculates the execution time and repeats the execution again on the database "tblenc_rc6" then calculates the execution time. Lastly repeats the execution again on the database on "tblenc_rea" then calculates the execution time. Then times for all the four cases will be compared and graphs for encryption and query process will be drawn for each user.

Database table for Users, tblusers:

id	Name	Password	Gen_key	Other_info	Utype
38	rajat	a872ed9b535	dhRR	Rajat	Student
39	payal	5fd572d6006	ayPd	Payal	Student
40	kajal	1cd2f0fc981	Adta	Kajal	Student
41	kdp	26329a7dd0	DK	Parate	Teacher

Table1: Database table, tblusers

We encrypted sixteten different fields in the table "tblenc_aes", "tblenc_rc6", and "tblenc_rea" with the proposed encryption algorithm REA and calculated elapsed time for each one shown in Table 2. Then, we calculated the averages of the elapsed times. We repeated this step on other encryption algorithms.

We decrypted the same sixteen different fields shown with decryption algorithms and calculated elapsed time for each one. We start executing the queries on these databases. Every query from the first to the fourth executes on the database "tblenc_plain" then calculates the execution time and repeats executes on the database "tblenc_aes" then calculates the execution and repeats the execution again on the database "tblenc_rc6 and tblenc_rea" then calculates the execution time Then, we calculated the averages of the elapsed times shown in Table 3.

V. Simulation Result

A comparison has been conducted for those encryption algorithms at encryption and decryption time. The encryption time is considered the time that an encryption algorithm takes to produce a ciphertext from plaintext. It indicates the speed of encryption. The decryption time is considered the time that decryption algorithm takes to produce a plaintext from ciphertext. Also, it indicates the speed of decryption.

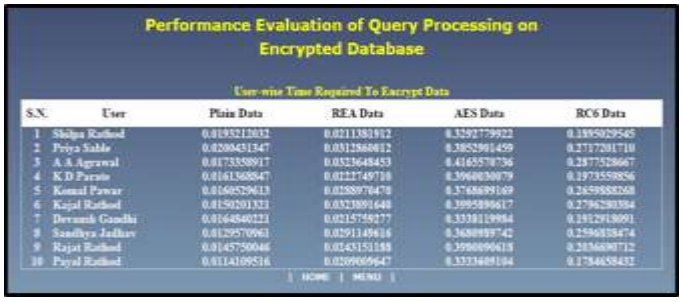

Table 2: Comparative elapsed times of encryption algorithms

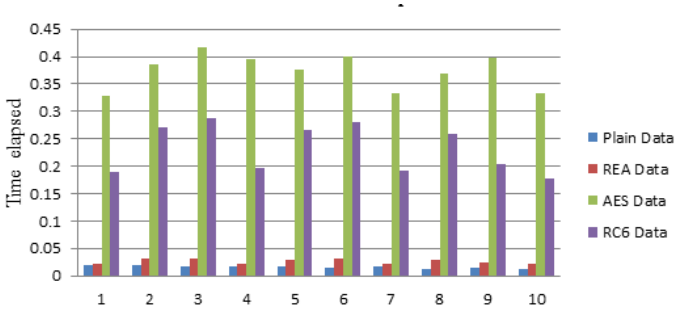

Figure 4: Time elapsed for encryption algorithms

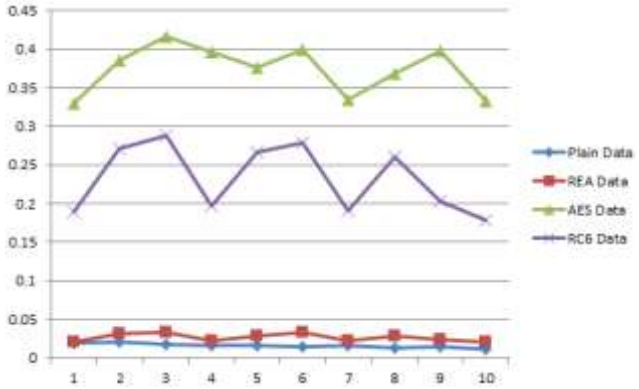

Figure 5: Time elapsed for encryption algorithms

Query	Plain	AES	RC6	REA
Query-1	0.003996	0.185551	0.152909	0.02883
Query-2	0.002388	0.205789	0.137024	0.039617
Query-3	0.002612	0.199909	0.132741	0.043172
Query-4	0.00399	0.202608	0.147289	0.032517

Table 3: Comparative elapsed times of retrieval algorithms.

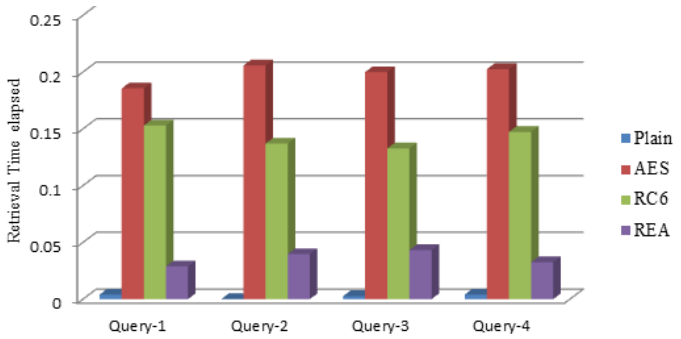

Figure 6: Time elapsed to retrieve data

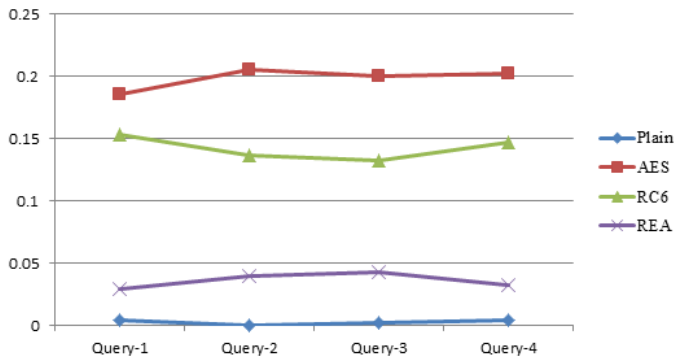

Figure 7: Time elapsed to retrieve data

Algorithm	Plain	REA	AES	RC6
Avg Time	0.02	0.03	0.44	0.26

Table 4: Average time elapsed for 20 users

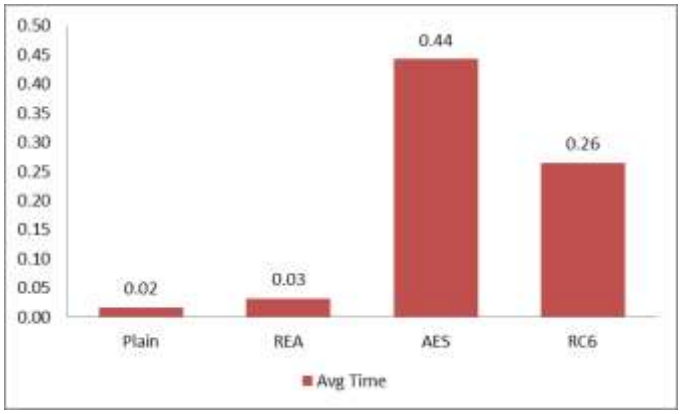

Figure 8: Average time elapsed

The results for this comparison are shown on Table 2 and Figure 4 and Figure 5 at the encryption time and Table 3 and Figure 6 and Figure 7 at the retrieval time. Also the average time elapsed for 20 users are evaluated for same number of queries on encrypted database. The result is shown in the Figure 8.

A first point: the database "tblenc_plain" takes less time than the other databases in terms of the query execution time. Because it has no encrypted fields so it takes less time.

A second point: the database "tblenc_aes" takes more time than other databases in terms of the query execution time.

A third point; the database "tblenc_rc6" is slower than the database "tblenc_plain" and faster than the database "tblenc_aes" in terms of the query execution time.

Finally; the database "tblenc_rea" is slower than the database "tblenc_plain" but fast as compared to other two database in terms of the query execution time. Overall, the results showed that REA has a very good performance compared to other algorithms.

All our experiments were done on laptop Pentium Core2Duo 2.0 GHz Intel Pentium processor with 1 MB cache memory, 3 GB of memory, and one Disk drive 320 GB. The Operating System which was used is Microsoft Windows 7 Ultimate. The results were executed based on the database MySQL, and the application developed in PHP and HTML.

VI. Conclusion

The aim of this project was to evaluate the performance of the query processing by using various algorithms like AES, RC6 and REA. We introduced an encryption algorithm, REA, restating its benefits and functions over other similar encryption algorithms. It limits the added time cost that is delay time for encryption and decryption so as to not degrade the performance. The performance measure of query processing conducted in terms of query execution time. The results of a set of experiments show the superiority of the algorithm REA over other encryption algorithms AES and RC6 with regards to the query execution time.

The proposed methodology shows the superiority of the encryption algorithm REA over other encryption algorithms AES and RC6. By using REA algorithm if we encrypt the database, the query performance of a system with regard to time is enhanced. REA can reduce the cost time of the operations and improve the performance. This is achieved by the encryption techniques used in REA algorithm, as there are no iterations used and variable key is used.

VII. Acknowledgment

First and foremost I want to thank my Guide Dr.C.A.Dhote for constant encouragement and noble guidance. With great pleasure and gratefulness, I extend my deep sense of gratitude to Prof.Dr.G.R.Bamnote, HOD, Computer Science & Engg. Deptt.for giving me an opportunity to accomplish my paper and to increase my knowledge. Lastly I wish to thank each and every person involved in making my dissertation successful.

References

[1] Michael L. Rupley, Jr., "Introduction to Query Processing and Optimization"

[2] Jinbiao Hou, "Research on Database Security of E-Commerce Based on Hybrid Encryption", *ISBN 978-952-5726-00-8 (Print), 978-952-5726-01-5 (CD-ROM). Proceedings of the 2009 International Symposium on Web Information S*

[3] Ayman Mousa, Elsayed Nigm, Sayed El-Rabaie, Osama Faragallah, "Query Processing Performance on Encrypted Databases by Using the REA Algorithm", *International Journal of Network Security, Vol.14, No.5, PP.280-288, Sept. 2012*

[4] Ayman Mousa, Elsayed Nigm, Sayed El-Rabaie, "Evaluating the Performance of Reverse Encryption Algorithm (REA) on the Databases", *The International Arab Journal of Information Technology, Vol. 10, No. 6, November 2013*

[5] Diaa Salama Abd Elminaam1, Hatem Mohamed Abdual Kader2, and Mohiy Mohamed Hadhoud, "Evaluating The Performance of Symmetric Encryption Algorithms", *International Journal of Network Security, Vol.10, No.3, PP.213-21. May 2010*

[6] Ayman Mousa, Elsayed Nigm, Osama Faragallah, Sayed El-Rabaie, "Security Analysis of Reverse Encryption Algorithm for Databases", *International Journal of Computer Applications (0975 – 8887) Volume 66– No.14, March 2013*

[7] Manish Sharma, Atul Chaudhary, Santosh Kumar, " Query Processing Performance and Searching over Encrypted Data by using an Efficient Algorithm", *International Journal of Computer Applications (0975 – 8887) Volume 62– No.10, January 2013*

[8] Ling Feng, "Experimental Evaluation of Query Processing on Encrypted Telemedicine Data", *978-1-4244-9166-7/10 $26.00 2010 IEEE*

[9] Gil-Ho Kim, Jong-Nam Kim, Gyeong-Yeon Cho, "*An improved RC6 algorithm with the same structure of encryption and decryption"*

[10] Sheetal Charbathia and Sandeep Sharma, "A Comparative Study of Rivest Cipher Algorithms", *International Journal of Information & Computation Technology. ISSN 0974-2239 Volume 4, Number 17 (2014), pp. 1831-1838*

[11] Ronald L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin, "The RC6 Block Cipher"

[12] T.Gunasundari1, Dr. K.Elangovan, "A Comparative Survey on Symmetric Key Encryption Algorithms", T.Gunasundari *et al*, International Journal of Computer Science and Mobile Applications, Vol.2 Issue. 2, February- 2014, pg. 78-83

[13] Diaa Salama Abdul Minaam, Hatem M. Abdual-Kader, and Mohiy Mohamed Hadhoud, "Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types", *International Journal of Network Security, Vol.11, No.2, PP.78{87, Sept. 2010*

[14] Database Security and Encryption A Survey Study

[15] Vijayalakshmi.G, Hema.S, Geethapriya.S, "Secure Data Aggregation & Query Processing in Wireless Sensor Networks using Enhanced Leach Protocol", International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-2, Issue-1, November 2013

[16] Ernesto Damiani, Sabrina De Capitani di Vimercati, Mario Finetti, "Implementation of a Storage Mechanism for Untrusted DBMSs**"**

[17] Stallings W., Cryptography and Network security Principles and practice, Prentice Hill-2005

[18] Nian Liu, Yajian ZhOU, Xinxin Niu, Yixian Yang, "Querying Encrypted Character Data in DAS Model", *2010 International Conference on Networking and Digital Society*

[19] P.Mohan Kumar, T.K.Das, DR.J.Vaideeswaran, "Survey on Semantic Caching and Query Processing in Databases", Proc. of the Second Intl. Conf. on Advances in Computer, Electronics and Electrical Engineering -- CEEE 2013 Copyright © Institute of Research Engineers and Doctors. All rights reserved. ISBN: 978-981-07-6260-5 doi:10.3850/ 978-981-07-6260-5_11

[20] Prof. S. S. Asole, Ms. S. M. Mundada, " A Survey on Securing Databases From Unauthorized Users", *NTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 2, ISSUE 4, APRIL 2013*

[21] G. Ramesh, R. Umarani, "Performance Analysis of Most Common Encryption Algorithms on Different Web Browsers", *I.J. Information Technology and Computer Science*, 2012, 12, 60-66 Published Online November 2012 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijitcs.2012.12.06

[22] 22.N.Bruno, L.Gravano and A.Marian. Evaluating Top-k Queries over web accessible databases. In proceedings of the 18th International conference on data engineering. SanJose, California, 2002

[23] Diaa Salama Abdul. Elminaam, Hatem M. Abdul Kader and Mohie M. Hadhoud, "Studying the Effects of Most Common Encryption Algorithms", *International Arab Journal of e-Technology, Vol. 2, No. 1, January 2011*

[24] Thenmozhi.C & Kishore Sonti, "Analyzing the performance of RC6 using Complex Vedic Multiplier", IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, issue 1, March, 2013 ISSN: 2320 – 8791

[25] Chong Hee Kim,"Improved Differential Fault Analysis on AES Key Schedule" IEEE Transaction on Information Forensics and Security, Vol. 7, No. 1, Feb 2012

[26] Irbid, Jordan, "A new approach for complex encrypting and decrypting data" International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.2, March 2013.

[27] Tingyuan Nie, Yansheng Li and Chuanwang Song, "Performance Evaluation for CAST and RC5 Encryption Algorithms", International Conference on Computing, Control and Industrial Engineering, IEEE, 2010

[28] Ratnmala Bhimanpallewar and Pravin Metkewar, "Aggregation Query Processing in P2P Networks", International Journal of Database Theory and Application Vol. 5, No. 3, September, 2012

[29] Ms. Mira K. Sadar, Mr. Pritish A.Tijare, Mr.Swapnil N.Sawalkar, "Database attacks and security: a review", International Journal of Advanced and Innovative Research (2278-7844) / # 203 / Volume 3 Issue 4

[30] Dr. Keshava Prasanna, Dr. Thungamani M., "Secured Query Processing in Wireless Sensor Networks", International Journal of Engineering Innovation & Research Volume 2, Issue 6, ISSN: 2277 – 5668

[31] Wang, S., Agrawal, D., Abbadi, A.E. A comprehensive framework for secure query processing on relational data in the cloud. Technical report, Department of Computer Science, UCSB (2010

[32] Priti V. Bhagat, Kaustubh S. Satpute, "Reverse Encryption Algorithm - A New Approach For Encryption", IOSR Journal of Computer Engineering (IOSR-JCE) ISSN: 2278-0661, ISBN: 2278-8727, PP: 57-62

[33] Website http://www.codeproject.com/Articles/2545/RC-encryption-and-decryption

[34] http://www.ukessays.com/essays/computer-science/ compressed-images-with-improved-encryption-computer-science-essay.php

[35] http://en.wikipedia.org/wiki/RC6_cipher