A Proposed Algorithm for Database Encryption and Decryption

Tarendra G. Rahangdale
2nd Sem. M.E, CSE
Sipna C.O.E.T. Amravati, India

Pritish A.Tijare
Dept.Computer Science and Engineering
Sipna C.O.E.T. Amravati, India

Swapnil N. Sawalkar
Dept.Computer Science and Engineering
Sipna C.O.E.T. Amravati,India

ABSTRACT
As the Computer System becomes popular for storing personal and precious data, need of data security goes its peak. For transmitting confidential information over the network, security is required so that it could not be accessed by illegitimate user. The database contains large amount of data that need to be secure. Cryptographic algorithms provide a way to secure data against the unauthorized access. Encryption is the process of encoding data so that its meaning is not obvious, decryption is reverse process that transform an encrypted massage back into original form. Encryption in database system is an important aspect for research, as efficient and secure algorithms are needed that provide the ability to query over encrypted database and allow optimized encryption and decryption of data. In this paper we observe an algorithm which encrypt and decrypt database over query fire quickly. This proposed algorithm will be simple and fast enough for most application which limits to the time and cost of encryption and decryption.

Keywords
Encryptions, Decryption, Information security, Database security, query processing.

1. INTRODUCTION
We knew that mostly the information security is provided by operating system security and physical security. But neither of these methods provides sufficient level of security to storing and processing the sensitive data. Cryptography is another important dimension of database security. The word cryptography comes from the Greek words for "secrete writing" and it consider as a science of that. It is the art of encoding data in such a format that is not easily decoded in its original format. It has a long history going back thousands of years. It is complementary to the access control because both of them can be used to guide the storage and access of confidential data in the database system. The encoded form of data in cryptography is known as cipher text. By performing various transformations cryptography renders the data unintelligible to unauthorized person. In [1, 2, 3, 4] database encryption mechanism enables following security.

i) Encryption mechanism can restrict the user from obtaining data in an unauthorized manner.

ii) Encryption mechanism prevents from leaking information in a database when storage mediums, such as CD-ROM, disk, and magnetic tape, are lost.

iii) Encryption mechanism can verify the authentic source of a data item.

However, the encrypted database becomes a challenge to efficient query fire. This implies that the system has to sacrifice the performance to obtain the security. For querying on encrypted database, we have to decrypt all the encrypted data. It is impractical because the cost of decryption for overall encrypted data is very expensive [5].

The purpose of this paper is to overview the following algorithm and remove a key removal problem associate it with.

In the cryptographic algorithm message to be encrypted, known as the plaintext, are transformed by a function that is parameterizes by a key. The output of the encryption process, known as the encrypted text or cipher text, is then transmitted on the network [6].

2. RELATED WORK
In [7] state a new encryption algorithm scheme (Chaotic Order Presenting Encryption (COPE)). It hides the sequence of the encrypted values by changing the sequence of buckets in the plaintext domain. COPE is secure against known plaintext attack. However, it can be used to perform many queries such as range and join queries. The overhead of range queries over encrypted database is higher than the overhead of range queries over plaintext database. In addition, it uses more than one key to change the order of the buckets and in some case that may result in duplicated values. Another drawback in COPE is the cost of encryption and decryption. That is because of the complexity to randomize the buckets and assign the right order within each bucket.

The bucketing approach [8, 9, 10, 11,12] is to divide the plaintext domain into many partitions (buckets). The encrypted database in the bucketing approach is augmented with index of attributes, thereby allowing query processing to some extent at the server without scarifying data privacy. In bucketing approach the encrypted database contains etuples and corresponding bucket-ids. In this scheme, executing a query fire over the encrypted database is based on the index of attributes. The outcome of this query is a superset of records that contains false positive tuples. These false hits must be eliminated in a post filtering process after etuples returned by the query which are decrypted. As only the bucket-id is used in a join operation, filtering can be complex. The projection operation is not used over the encryption database, because a row level encryption is implemented.

3. THE STUDIED ENCRYPTION ALGORITHM
In Today's world it requires that the encryption/decryption algorithm should be relatively simple and efficient. We studied this algorithm because it fulfills the requirements.

This algorithm is symmetric stream cipher that uses a single key of any variable length. This algorithm has same encipherment and decipherment stages, except the two stages:

i) Adding the key to plaintext in encryption and removing it while decryption.

ii) Performing divide operation on the plaintext by 4 in encryption and multiply the cipher text by 4 at the time of decryption.

The reason behind the operation of diving the text by 4 is to narrow the range domain of the ASCII code table at converting the text. The details about the working of the proposed algorithm are given below.

3.1 The Proposed Encryption Algorithm

Following are the steps of encryption algorithm.

Step 1: Input any length of plaintext and the key.

Step 2: Add the key to the plaintext at any place and save position of each element of key before plaintext.

Step 3: Convert the result of step2 into its ASCII code.

Step 4: Convert the ASCII code of step3 into each 8 bit binary data.

Step 5: Reverse the previous binary data from step4.

Step 6: Take One's compliment of data of step5.

Step 7: Collect each 8 bits from the above binary data and obtain its ASCII value.

Step 8: Divide each ASCII value of above step by 4.

Step 9: Obtain the ASCII code of quotient of above division and put it as one character.

Step 10: Obtain the remainder of the division and put it as a second character.

Step 11: Resulting data is our cipher text.

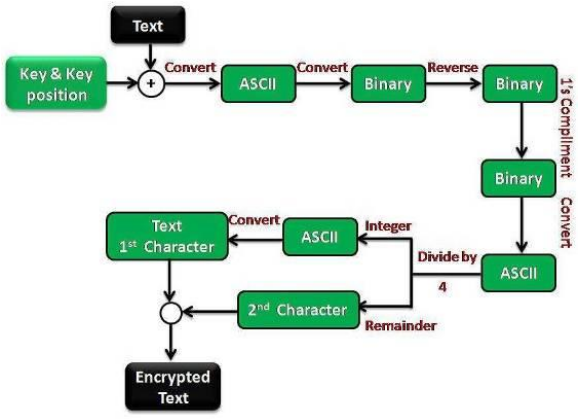

Figure 1: Step of Proposed Encryption Algorithm

3.2 The Proposed Decryption Algorithm

Following are the steps of decryption algorithm.

Step 1: Input the cipher text and the key.

Step 2: Loop on the cipher text to get ASCII code of characters.

Step 3: Multiply the ASCII code of Step 2 with 4 and add the next digit (Remainder).

Step 4: Convert the ASCII code of Step 3 to 8 bit binary data.

Step 5: Take the One's compliment of the binary data.

Step 6: Reverse the binary data of Step 5.

Step 7: Arrange the binary data 8 bit each and obtain its ASCII code.

Step 8: Convert the ASCII code to its respective ASCII character code.

Step 9: Remove the key from result of Step 8 using the length and sequence of key element.

Step 10: Resulting the original plaintext.

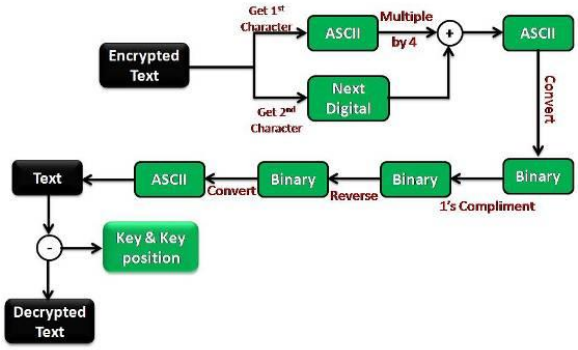

Figure 2: Step of Proposed Decryption Algorithm

Example:

To understand the actual working of the proposed encryption algorithm, we have taken an example.

Consider the text is: "Hey!"

And the key is: "123".

Following are the steps involving in performing the encryption algorithm.

Step 1: Input text as "Hey!" and key as "123".

Step 2: Add the key to the plaintext at any place.

"Hey!123" or it may be any other combination as "123Hey!", "H1e2y3!", and get the position of each element of key in plaintext and put it before plaintext i.e. In "Hey!123" it will "456Hey!123".

Step3: Convert the "456Hey!123" in ASCII code and then in its binary form.

4→52➔00110100, 5→53➔00110101

6→54➔00110110, H→72➔01001000

e→101➔01100101,y→121➔01111001

!→33➔00100001, 1→49➔00110001

2→50➔00110010, 3→51➔00110011

So the binary data will be as

00110100 00110101 00110110 01001000 01100101 01111001 00100001 00110001 00110010 00110011

Step4: Reverse the previous binary data.

11001100 01001100 10001100 10000100 10011110 10100110 00010010 01101100 10101100 00101100

Step 5: Take One's compliment of above binary data.

00110011 10110011 01110011 01111011 01100001 01011001 11101101 10010011 01010011 11010011

Step 6: Collect each 8 bits from the above binary data and obtain its ASCII code.

00110011➔51, 10110011➔179,

01110011➔115, 01111011➔123, 01100001➔97, 01011001➔89, 11101101➔237, 10010011➔147, 01010011➔83, 11010011➔211

Step 7: Divide each ASCII code value of above step by 4.Obtain the quotient of above division.

Table 1: Getting ASCII character and remainder for cipher text.

	Quotient	Character ASCII	Remainder
51/4	= 12	♀	3
179/4	= 44	,	3
115/4	= 28	␣	3
123/4	= 30	▲	3
97/4	= 24	↑	1
89/4	= 22	_	1
237/4	= 59	;	1
147/4	= 36	$	3
83/4	= 20	¶	3
211/4	= 52	4	3

Step 8: For generating cipher text arrange the Character ASCII as first character and its remainder as second character. The encrypted text is:

♀3,3␣3▲3↑1_1;1 $1¶343

3.3 Proposed Decryption Algorithm

Following are the steps involving in performing the encryption algorithm.

Step 1: Take the input as encrypted text and the key for farther use.

Step 2: Loop on the cipher text to get ASCII code of characters.

♀ →12 , →44 ␣ →28

▲→30 ↑→24 _→22

; →59 $ →36 ¶→20

4 →52

Step 3: Multiply it with 4 and add the next digit (Remainder).

Table 2: Get ASCII code of character

	ASCII code *4	Remainder	(ASCII)
♀ →	12*4	+ 3	= 51
, →	44*4	+ 3	= 179
␣ →	28*4	+ 3	= 115
▲→	30*4	+ 3	= 123
↑→	24*4	+ 1	= 97
_ →	22*4	+ 1	= 89
; →	59*4	+ 1	= 237
$ →	36*4	+ 3	= 147
¶ →	20*4	+ 3	= 83
4 →	52*4	+ 3	= 211

Step 4: Convert the ASCII code value of Step 3 to 8 bit binary data.

51➔00110011, 179➔10110011,

115➔01110011, 123➔01111011 97➔01100001, 89➔01011001, 237➔11101101, 147➔10010011, 83➔01010011, 211➔11010011

00110011 10110011 01110011 01111011 01100001 01011001 11101101 10010011 01010011 11010011

Step 5: Take the One's compliment of the binary data.

11001100 01001100 10001100 10000100 10011110 10100110 00010010 01101100 10101100 00101100

Step 6: Reverse the above binary data.

00110100 00110101 00110110 01001000 01100101 01111001 00100001 00110001 00110010 00110011

Step 7: Obtain the ASCII code of binary data and then convert to ASCII character code (As data is already in 8 bit form).

00110100➔52→4, 00110101➔53→5

00110110➔54→6, 01001000➔72→H

01100101➔101→e, 01111001➔121→y

00100001➔33→!, 00110001➔49→1

00110010➔50→2, 00110011➔51→3

The text is: 456Hey!123

Step 8: Remove the key from resulting text using position of element of key.

The decrypted plaintext is: "Hey!"

4. OUR CONTRIBUTION

In the studied algorithm there is a problem related to key removal [13], in overview we try to remove that drawback by placing the position of key in front of plaintext. That key position will help to get the correct plaintext while decryption.

5. CONCLUSION

Cryptography is an important process of securing our important data. In this paper, we studied an encryption algorithm. This algorithm is fast and simple for most application. This proposed algorithm will reduces the cost and time complexity of the encryption/decryption operation so that system performance will improve.

6. REFERENCES

[1] H.Brown, Considerations Database Management System Encryption Security Solution, A Research Report presented to The Department of Computer Science at the University of Cape Town, 2003.

[2] G. Davida, D. L. Wells, and J. B. Kam, "A database Encryption system with subkeys," ACM Transactions on Database Systems, vol. 6, no. 2, pp. 312–328, 1981.

[3] H. Hacig¨um¨us, B. Iyer, and S. Mehrotra, "Providing database as a service," in Proceedings of ICDE, pp. 29–38, 2002.

[4] J. He and M. Wang, "Cryptography and relational database management system," IDEAS, pp. 273–284, 2001.

[5] Oracle, Oracle9i Database Security for eBusiness, An Oracle White Paper, June 2001.

[6] http://en.wikipedia.org/wiki/Encryption

[7] S. Lee, T. 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, pp. 207–217, 2009.

[8] Ceselli, E. Damiani, S. D. C. D. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati, "Modeling and assessing inference exposure in encrypted databases," ACM Transactions on Information System Security, vol. 8, no. 1, pp. 119–152, 2005.

[9] E. Damiani, S. D. C. D. Vimercati, M. Finetti, S. Paraboschi, P. Samarati, and S. Jajodia, "Implementation of a storage mechanism for untrusted dbmss," IEE Security in Storage Workshop 2003, pp. 38-46, 2003.

[10] E. Damiani, S. D. C. D. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Metadata management in outsourced encrypted databases," in The Second VLDB Workshop on Secure Data Management, Lecture Notes in Computer Science, pp. 16–32, Springer, 2005.

[11] H. Hacig¨um¨us, B. R. Iyer, and S. Mehrotra, "Ensuring the integrity of encrypted databases in the database-as-a-service model," DBSec 17th Annual Working Conference on Data and Application Security, pp. 61–74, Kluwer, 2003.

[12] Q. Tang and D. Ji "Verifiable attribute based encryption," International Journal of Network Security, vol. 10, no. 2, pp. 114-120, Mar. 2010.

[13] Assessing performance of encrypted databases under query processing with the REA Algorithm Volume 2, Issue 1, January 2014, ISSN: 2321-7782.