The International Journal Of Engineering And Science (IJES)
|| Volume || 2 ||Issue|| 12 || Pages || 92-98 || 2013 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Throughput Comparison Results of Proposed Algorithm with Existing Algorithm

[1]Naveen Kolhe, [2]Nikhat Raza
[1]*Department of Computer Science, BIST, Bhopal, India*
[2]*Department of Computer Science, BIST, Bhopal, India*

----------**ABSTRACT**----------
It is known that encryption is one of the strongest security solutions for database, but developing a database encryption strategy must take many factors into consideration. Basically encryption should be performed where data is originating. This paper examines the various issues of implementing database encryption and makes recommendations. This paper is concluding with a benchmark using the proposed design criteria like proposed encryption/decryption algorithm. Presented results are showing the performance of the proposed algorithm.

Keywords *- Algorithm, Cryptography, Database, Decryption, Encryption,.*

Date of Submission: 02 December 2013 Date of Publication: 15 December 2013

I. INTRODUCTION

This Paper titled "Throughput Comparison Results of Proposed Algorithm with Existing Algorithm" enhances a new parameter called throughput results to the other parameters of cryptography algorithm proposed by analyzing the principle of the encryption/decryption algorithm based on symmetric key cryptography. Moreover, the security and performance of the proposed algorithm is also estimated [1] . The symmetric key [2] approves the effectiveness of the proposed method, and it shows advantages of large key space and high-level security. The cipher text generated by this method can be vary in size as the plaintext and is suitable for practical use in the secure transmission of confidential information over the Internet. The objective of proposed research is to propose a general database [3] cryptography algorithm, without explaining the fixed symmetric or asymmetric encryption algorithms used in that algorithm. The proposed encryption algorithms affect the performance of execution and security analysis. Proposed research will investigate each issue in detail. Our proposed algorithm does rely on a specific symmetric encryption algorithm; proposed symmetric algorithm with the appropriate key size can be used. Also, it is possible to apply different encryption algorithms on different sides. For example, the user could use algorithm A, the other user could apply algorithm B, and the another user could do it with algorithm C. This provides a flexible and more secure encrypting environment. The concept of database security and the word cryptography might be intimidating and complicated. The objective of the report is to develop a tool that mediates the user and the operations to achieve database security goals. A platform independent tool with user-friendly graphical user interface, using already existing techniques and algorithms for cryptographic operations will be resulting product. People need to use the cryptographic operations in order to keep the personal sensitive information files to avert from foreigners in consideration of the security goals. The operations include bunch of algorithms to protect the attacks of foreigners to reach and read personal files that is located in personal computer or the owner would like to send somewhere. Cryptographic operations consist of encryption and decryption techniques in computer and computer networking. In effort to keep information's in safety such as banking account information's or to provide file transaction without any problem such as password sharing caused such a security methods.

II. PROPOSED WORK

"Throughput Comparison Results of Proposed Algorithm with Existing Algorithm" shows throughput results of proposed symmetric key encryption/decryption algorithm which is improved version of existing algorithm like AES. The proposed work overcomes the security limitations of existing algorithm by adding a new level of confusion thus resulting in a strong cryptographic algorithm. The proposed algorithm also ensures that encryption and decryption time is similar to that of the existing system. The architecture in the Fig 1 and 2 below shows the process of encryption and decryption:

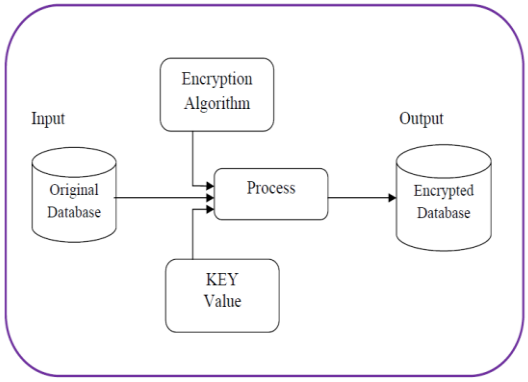


Fig1: Architecture diagram for encryption

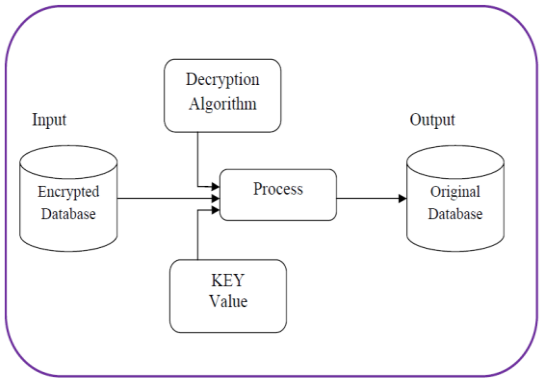


Fig2: Architecture diagram for decryption

Proposed Concept: In proposed system, whose basic architecture is shown in Fig 3, apply encryption/decryption process accumulatively while data moves from/to the database. The purpose of such design is to implement encrypted storage satisfying data confidentiality, privacy and integrity. In addition, it makes data usable for the server side, and makes it possible for the database to be joined to other databases by the coordinator. The encryption operation in proposed system is based on mathematically operation. Such operation is needed to draw a full map of the encryption process.

Following the convention, *E* denotes the encryption function, and *D* denotes the decryption function. The general approach to encryption is the field level; that is, sensitive fields need to be encrypted to protect the information from inside or outside attack. Data is encrypted using a symmetric encryption algorithm. After the process, each side encrypt the data its own. For each database schema $S(R_1; :::;R_n)$, where R_1 to R_n are relations created on the database, R_E name is encrypted database using the secret key. For database security where information which is stored in the database is encrypted by using proposed symmetric cryptography algorithm which will be based on block cipher concept where each information of database will treat as a block of equal length and then each block will encrypt using a special mathematical set of functions known as Key with the help of proposed encryption algorithm.

During encryption or decryption same key will use due to symmetric nature. In the proposed algorithm 128 bits of key length is used so that security of proposed algorithm will be very high. In Fig 3 databases will execute with proposed encryption algorithm and proposed encryption algorithm will call to proposed key to produce encrypted database. In reverse encrypted database will execute with proposed decryption algorithm and this proposed decryption algorithm will call same proposed key to produce original database.

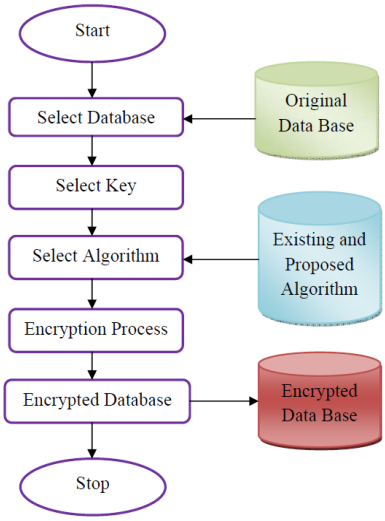


Fig 3: Proposed System Architecture

Following example is showing work of the proposed research. In the employee example, the database is encrypted based on the data classification proposed in this research. Table 1 shows an unencrypted employee database and Table 2 shows the encrypted database.

Table-1: Unencrypted Database of Employee as an input

S. No	Eid	Ename	Edoj	Edob	Esalary	Eaddress
1	101	ram	12-12-2000	12-12-1980	5000	Bhopal
2	102	Raj	18-1-2003	3-5-1983	5000	Indore
3	103	rajesh	1-12-200	13-6-1978	4000	jabalpur

Table-2: encrypted Database of Employee as an output

S. No	Eid	Ename	Edoj	Edob	Esalary	Eaddress
1	@s#	*&”	“:!\s3?q	!2sd%$#m	3%!nd*	A|.ls,#@
2	%)!	&)*	!#aw)”|2<’	Nd*&@^>	*Nk3^5	@.3d,%&
3	(*^	*&^%$%	-wl)*>&2s	3!}sk#ms	!&*?|	*34hs>?1

The Size of the encrypted Table 2 can be increase; it will depend upon the encryption algorithm, key size and other parameters.

Block Diagram of Proposed Encryption: Fig 4 is showing basic block diagram of proposed encryption algorithm. In this user select 128 bits key length and select database. Then proposed encryption algorithm executed after completing execution encrypted data one again stored in the same database which is known encrypted database.

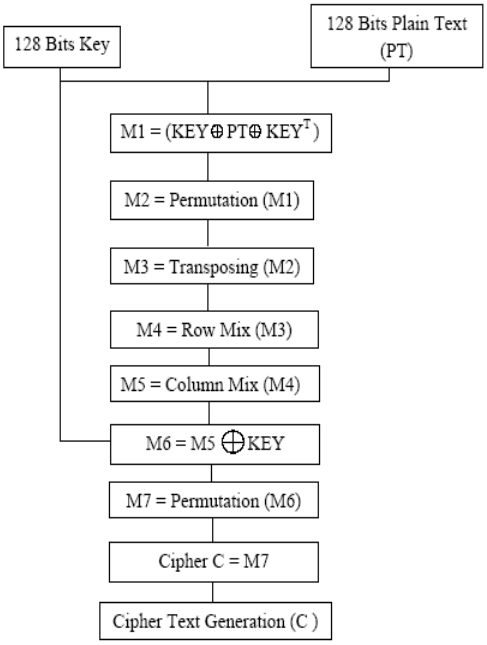


Fig 4: Block Diagram of Proposed Encryption

Proposed Encryption Algorithm:

1. Input Key (KEY) of 16 byte or 128 bits length.
2. Arrange Key in matrix of 4 X 4.
3. Calculate Key Transpose (Key^T)
4. Select plain text (PT) of 16 byte or 128 bits from data base.
5. Arrange plain text in matrix of 4 X 4.
6. Apply XOR operation between key, plain text and key transpose. Out put of this step will be message (M1).
7. Perform permutation function on message (M1. Out put of this step will be message (M2).
8. Perform transpose function on message (M2). Out put of this will be message (M3).
9. Perform row mixing function on message on (M3). Out put of this step will be message (M4).
10. Perform column mixing function on message (M4). Output of this step will be message (M5).
11. Apply XOR operation between message (M5)and key (KEY) value. Out put of this step will be message (M6).
12. Perform permutation function on message (M6). Out put of this step will be message (M7).
13. Message (M7) is the final cipher text.
14. Exit.

Block Diagram of Proposed Decryption: Fig 5 is showing basic block diagram of proposed decryption algorithm. In this user select 128 bits key length and select encrypted database. Then proposed decryption algorithm executed after completing execution original data one again stored in the same database which is known original database.

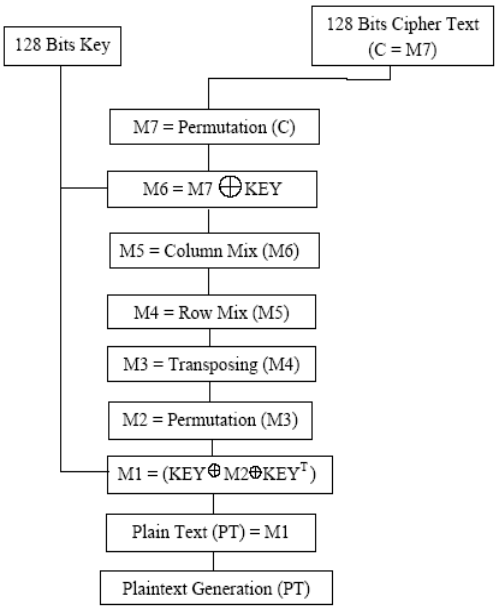


Fig 5: Block Diagram of Proposed Decryption

Proposed Decryption Algorithm:

1. Input Key (KEY) of 16 byte or 128 bits length.
2. Arrange Key in matrix of 4 X 4.
3. Calculate Key Transpose (Key^T)
4. Select cipher text (CT) of 16 byte or 128 bits from data base.
5. Arrange cipher text (CT) in matrix of 4 X 4.
6. Perform reverse permutation function on message (CT). Out put of this step will be message (M1).
7. Apply XOR operation between message (M1)and key (KEY) value. Out put of this step will be message (M2).
8. Perform reverse column mixing function on message (M2). Output of this step will be message (M3).
9. Perform reverse row mixing function on message on (M3). Out put of this step will be message (M4).
10. Perform transpose function on message (M4). Out put of this will be message (M5).
11. Perform reverse permutation function on message (M5). Out put of this step will be message (M6).
12. Apply XOR operation between key, message (M6) and key transpose. Out put of this step will be message (M7).
13. Message (M7) is the original plain text.
14. Exit.

Proposed System Attribute:

Reliability and Fault Tolerance: The Proposed Software source code should be reliable and of enterprise quality.
Availability: The proposed software will be placed in public domain and it will be unpatented in all countries. Anyone would freely be able to use the proposed software. The specification, source code and test data for the proposed software would be available to anyone wishing to compare the algorithms, in accordance with country specific export laws.
Security: The proposed software will contain strong cryptographic algorithm, so even if it is created, maintained and distributed from liberal countries (where it is legal to do this), it falls under certain export/import and/or use restrictions in some other parts of the world. Please remember that export/import and/or use of strong cryptography software, providing cryptography hooks or even just communicating technical details about cryptography software is illegal in some parts of the world. So, when you import the reference implementation to your country, re-distribute it from there or even just email technical suggestions or even source patches to the author or other people you are strongly advised to pay close attention to any export/import and/or use laws which apply to you. Anyone associated with this project are not liable for any violations you make here.
Accessibility: The accessibility of our product is not dependent to any extra license. Propose software is a freeware product. Thus, every person is able to execute the program
on any platform.
Correctness and Consistency: The specification of proposed software will be correct and consistent.
Portability: The proposed software will be in C# Dot Net programming language (and will execute on any system where its compiler has been ported to). The portability of other software implementations will depend on portability of chosen programming language.
Performance: The requirement of the computers is not important for our product. It may be important for big sized files. Otherwise, performance is expected fast enough. The system shall react the user operations immediately.

III. RESULTS

Dot Net implementation is done to present an evaluation system. For encryption time and decryption time of the existing algorithm with this proposed algorithm, it is necessary to describe the detailed evaluation method, as illustrated in Fig 6. Here only one evaluating mode is taken to find whether the key and the plain text have impact on time consuming of cryptographic algorithms, DDSK (different size of database in the same key).

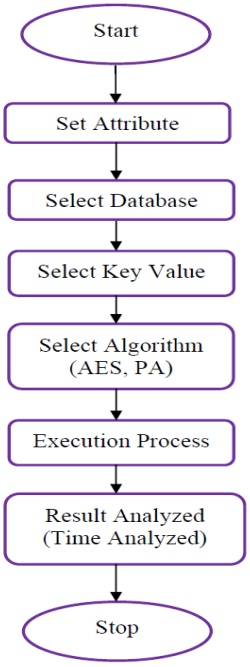


Fig 6: Results Evolution Model

For this experiment, a laptop Pentium® Dual-Core CPU T4400 @2.20Ghz and 32-bit Operating System is used, in which performance data is collected. In the experiments, the laptop encrypts a various size of database and new parameter Throughput is used to calculate performance of the existing algorithm as well as Proposed Algorithm (PA). The proposed system approximately 100 times has run. In each time, same database are respectively encrypted by Existing Algorithm and Proposed Algorithm (PA) by copying them. Finally, the outputs of the proposed system are execution time and decryption time, and calculated in numeric form.

Throughput: Throughput of the any algorithm can be calculated by the equation (1). It is the ratio of total size of database divide by total execution time during encryption/decryption.

Throughput = Total Size of Encrypted Database / Total Execution Time in Encryption ……..(1)

Table 1 is showing throughput comparison between AES and Proposed algorithm (PA). In this Table there are different size (21437/22080/28230) Bytes of dataset have used to calculate throughput.

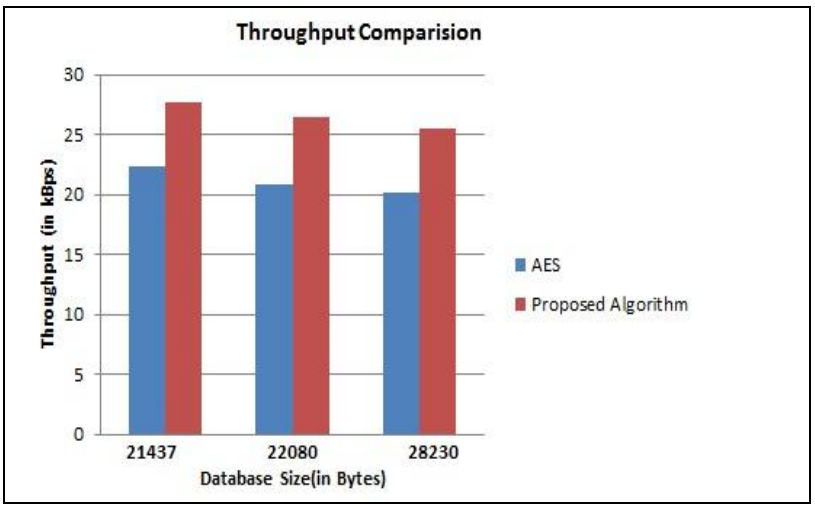
Throughput of the AES Algorithm is 22.39kBps for 21437 Bytes of Database.
Throughput of the AES Algorithm is 20.82kBps for 22080 Bytes of Database.
Throughput of the AES Algorithm is 20.23kBps for 28230 Bytes of Database.

Throughput of the Proposed Algorithm (PA) is 27.72kBps for 21437 Bytes of Database.
Throughput of the Proposed Algorithm (PA) is 26.55kBps for 22080 Bytes of Database.
Throughput of the Proposed Algorithm (PA) is 25.51kBps for 28230 Bytes of Database.

Table 3: Throughput Comparisons between Proposed and Existing Algorithm.

Database Size(in Bytes)	AES	Proposed Algorithm
	Throughput in kBps (Approx.)	
21437	22.39	27.72
22080	20.82	26.55
28230	20.23	25.51

The graph 1 is drawn from the Table 3 to reveal it. In this graph the key length of existing encryption algorithm like AES is 128-bit, which is equal as compare the proposed algorithm (PA).



Graph 1: Throughput Comparison between AES and Proposed Algorithm

Experimental result is shown in Table 3. Presented result shows the superiority of Proposed Algorithm (PA) as compare existing algorithms in terms of the execution time and new parameter i.e. Throughput comparison which is the objective of the research. Another point can be noticed here that Proposed Algorithm has good performance in terms of power consumption.

IV. CONCLUSION

In this section, throughput performance of the proposed encryption scheme is analyzed in detail. We discuss the security analysis of the proposed encryption scheme including some important ones like the complexity of time and space. However, it is a difficult problem to evaluate the specific algorithm, they must consider many factors: security, the features of algorithm, the complexity of time and space, etc, so research on the time-consuming of algorithm is one of the important aspects. In the past evaluating time-consuming of algorithm usually through comparing its time complexity, while in this research we are proposing a new evaluation model and two evaluation modes to measure the time-consuming of these cryptographic algorithms and our proposed algorithm.

REFERENCES

[1] Naveen Kolhe, Nikhat Raza and P. S. Patheja, A New Approach of Cryptography for Database Security , *International Journal of Emerging Technology and Advance Engineering, (ISSN 2250-2459*), Issue 2, February 2013.

[2] Stallings.W, Cryptography and Network Security, *Prentice Hall*, 4th Ed, 2005.

[3] Hasan Kadhem, Toshiyuki Amagasa and Hiroyuki Kitagawa "A Novel Framework for Database Security based on Mixed Cryptography" published in *Fourth International Conference on Internet and Web Applications and Services 2009 IEEE.*