A Comparative Analysis of Encryption Algorithms for Better Utilization

Anuj Kumar
M.tech(IT)
ASET
Amity University Noida

Sapna Sinha
Astt. Professor
ASET
Amity University Noida

Rahul Chaudhary
M.tech(CSE)
ASET
Amity University Noida

ABSTRACT
Cryptographic algorithms provide security against attacks during encryption of data. However, they are computationally intensive application and consume large amount of CPU time [1] and space at time of encryption. The goal of this paper is to compare the different encryption algorithm and to find space complexity of the encrypted and decrypted data by using complexities of encryption algorithm. . In this paper provide comparison between five most widely used algorithms. Based on following experimental it can be seen that TDES in general perform better than other algorithms. In this, find that how these algorithms better utilize for improving performances of algorithms in terms of space complexity.

General Terms
Algorithms

Key words
Cryptographic algorithm, encryption, block cipher, stream cipher, space complexity

INTRODUCTION
Encryption is basically hiding of data while being transmitted or stored [2].The encryption process consists of an algorithm and a key. The key controls the algorithm. The objective is to design an encryption technique so that it would be very difficult or impossible for an authorized party to understand the content of the cipher text. A use can recover the original message only by decrypting the cipher text using security key. Depending upon the secret key used, the algorithm will produce a different output. If the secret key changes then the output of the algorithm also changes. A key is used for performing encryption and decryption. Key is a specific number (usually large one) which is usually used by the algorithms and its calculation [2]. There are two types of encryption algorithms. These are discussed in detail below-
1. Symmetric key encryption
2. Asymmetric key encryption

1. SYMMETRIC KEY ENCRYPTION
In symmetric key encryption algorithm can be used only one secret key for encrypting and decrypting data.
While using symmetric key cryptographic encryption algorithms, key can be calculated from decryption key and also their vice versa. When implementing symmetric key encryption it can be very efficient, due to this user cannot face any significant time delays during encrypting and decrypting data. It provide a degree of authentication i.e. data cannot be decrypted through other key. It is only beneficial for the user if encryption key is kept secret. The authority of symmetric key encryption depends upon size of the key [1]. Symmetric key encryption algorithm is of two types-
1.1 Block cipher
1.2 Stream cipher

1.1 Block cipher
A block cipher can operates on block of data. Block cipher algorithms that permute N-bit block of plaintext data encrypted any other [2]. In this algorithm breaks into block and perform operation on each block independently. It uses blocks of 8 or 16 bytes long. Security of block cipher is basically depending upon the encryption function. Software implementation of block cipher runs faster than software implementation of stream cipher. Error transmitting in one block generally does not affect other block. The data contains in each block is encrypted independently, using the same key, identical plaintext blocks produce identical cipher text blocks. Suppose that plain text is 227 byte long and the cipher text you are using operates on 16-byte blocks. Algorithm takes the first 16-bytes of data, encrypts them using the key table. Algorithm provides 16-byte of cipher text. After first block, algorithm takes next block. The key table doesn't change from block to block.
Plain text= 227 bytes
Block size= 16 bytes
= 227÷7
= 14 blocks plus 3 bytes
Algorithm encrypts 14 bytes and 3 bytes remain. For encryption last 3 bytes data padding is used. Extra bytes are added to make the last block size of 16 bytes. Whoever decrypts the cipher text must be able to recognize the padding. One problem with block ciphers is that if the same block of plain text appears in two places, it encrypts to the same cipher text. To avoid having these kinds of copies in the cipher text, feedback modes are used. Cipher block chaining is not containing the extra information that acquires bit space, so every bit in the block is part of the massage. Before plain text is enciphered, that block is XORed with preceding cipher text block. It requires an initialization vector to XOR the initial plain text block in addition of a key. For decrypting the data, copy a block of cipher text, decrypt it and XOR the result with the preceding block of cipher text. Up to now, there have been number of research articles pointing out the performance of the compared algorithms [2]. Taking E_k to be the decipherment algorithm with key and initializing vector is I, technique used in this algorithm is-

$$C_0 = E_k(m_0 \oplus I)$$

$$C_i = E_k(m_i \oplus C_{i-1} \text{ For i > 0}$$

There are different block cipher algorithm-
1. One time pad
2. IDEA
3. Blowfish

4. RC2
5. Serpen
6. CAST-5
7. RC6

1.2 Stream cipher

Designing of these algorithms to accept a crypto key and a stream of plain text to produce a stream of cipher text. Stream cipher comprises of two main components: a mixing function and a key stream [2]. Mixing function is usually exactly an XOR function, whereas key stream generator is the main unit in stream cipher encryption [1, 3]. Stream cipher basically operates on small units of plain text. It is faster than block cipher. Stream cipher produces the input element continuously producing one output at a time. It uses fewer amounts of codes and key is uses only once. Many stream cipher algorithms are used for hardware implementation. Stream cipher encrypts smaller block of data, typically bits or bytes. A key stream generator outputs a stream of bits K1, K2,K3….Ki. This key stream is XORed with a stream of plain bits p1, P2,P3….Pi to produce the stream of cipher text bits.

$$C_i = P_i \oplus K_i$$

.At the description end, the cipher text bits are XORed with an identical key stream to recover the plain text bits.

$$P_i = C_i \oplus K_i$$

The system security depend s entirely on the inside of keys team generator. There are different stream cipher algorithms-

1. Salasa2
2. HC-125
3. VMP
4. RC4
5. HC25
6. Grain

2. Asymmetric key encryption

Asymmetric key encryption algorithm also called public key encryption algorithm. It is used in message authentication and key distribution. These algorithms are based on mathematical functions. It uses two separate keys i.e. encryption key and decryption both are different and the decryption key could not be derived from the encryption key. Only the authorized person can be able to decrypt the cipher text through his own private key [4]. Following steps are required for this algorithm.

1. Each end system in a computer network generates a pair of keys to be used for encryption and decryption of messages that I will receive.
2. Each system publishes its encryption key i.e. this is public key. The companion key is kept private.
3. If ‘X1’ wishes to send a message to ‘Y1’, it encrypts the message using Y1’s public key.

After ‘Y1’ receives the message, it decrypts the message by using Y1’s private key. The public key is accessed to all participants and private key us generated locally by each participant.

System controls its private key. At any time, a system can change its private key. Figure 1 shows the process of public key algorithm. A message from source which is in a plain text, X=(X1, X2, X3… X_m). The message is intended for destination which generates a related pair of keys a public key Ku_b , and a private key KR_b . Private Key is secret key and known only to Y1. With the message X and encryption x and encryption key Ku_b as input, X1 forms the cipher text.

Y= (Y1, Y2, Y3…. Y_n)

$$Y = E_{Kub}(X)$$

The receiver, in possession of the matching private key is able to invert the transformation.

$$Y = D_{KRB}(Y)$$

An opponent, observing y and access to public key (K_{ub}), but not having access to private key (K_{Rb}), must attempt to recover X. It is assumed that the opponent does have knowledge of the encryption (E) and decryption algorithms (D). Public key cryptography requires each user to have two keys: A public key used by anyone for encrypting messages to be sent to that user and a private key, which the user need to decrypting messages.

There is different asymmetric key encryption algorithms-

1. RSA encryption algorithm
2. Diffie–Hellman key exchange
3. Digital signature algorithm
4. ElGamal

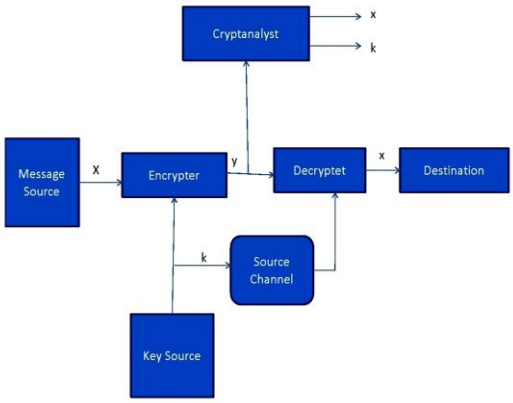

Fig1: Model of cryptographic system for Encryption and Decryption

3. IMPLEMENTED ALGORITHMS

These are following encryption algorithm that are chosen for the implementation-

3.1 DES
3.2 TDES
3.3 RSA
3.4 Blowfish
3.5 XOR

3.1 DES encryption

In may 1973, NIST (then NBS) called for possible encryption algorithms for use in unclassified adopted encryption algorithm and is many standard around the world (e.g. Australian standards AS28055-1985) [5]. The plaintext blocks of data in and put through an initial permutation.

1. Put plaintext K= {}
2. Divide plaintext K into n 64-bit block
3. Repeat for each block for i=0 to n-1
4. Performed calculation of initial permutation
5. After that divided into two parts
6. Po= left side sub part
7. Qo= right side sub part

8. round I has inputs Pi-1, Qi-1
9. Output of it will be
   Pi= Qi-1,
   Qi=Pi-1 XOR f(Qi-1,Mi)
10. For ith round Mi is the sub key where $1 \leq i \leq 16$
11. After completion of round 16, interchange Lo and Ro// which conforms decryption algorithm has same structure as encryption algorithm
12. At last, compute IP^{-1}
13. The output will be ciphertext i.e. output=ciphertext

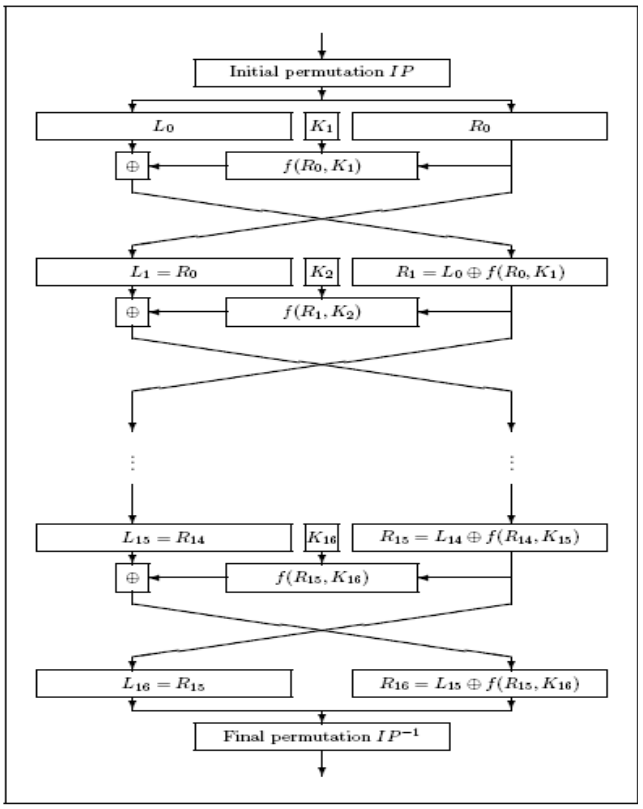

Fig2: DES Encryption [7]

3.2 TDES encryption

Triple DES is one of the other modes of encryption and decryption. It requires three 64-bit keys and having overall key length is 192 bits. TDES is a proposal based on the Existing DES, and was standardized in ANSI X9.17 & ISO 8732 and in PEM for key management [5]. The procedure for encryption is exactly the same as regular DES, but repeat it three times. The data is encrypted with the key (K1), decrypted with the second key (K2), and finally encrypted again with the third key (K3). Triple DES with three keys is required quiet extensively in many products including PGP and S/MIME. Brute force search impossible on triple DES. Meet-in-middle attacks needs 256 Plaintext-Cipher text pairs per key. Cipher text is produced as C=Ek3 [Dk2 [Ek1 [P]]].

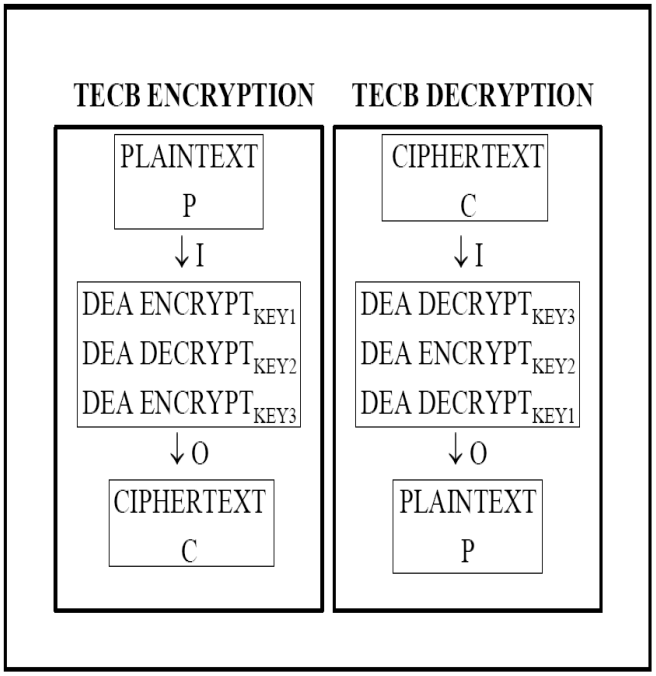

Fig3: TDES Encryption/Decryption [7]

3.3 RSA encryption

The RSA algorithm is developed in 1977 by Rivest, Shamir, Adleman (RSA) at MPT. It has been widely used for many years
on the internet for security and authentication in many applications including credit card payments, email and remote login sessions [6]. RSA algorithm is public key encryption type algorithm. In this algorithm, one user (party) uses a public key and other user uses a secret key (private key) key. In the RSA algorithm each station independently and randomly chooses two large primes p and q number, and multiplies them to produce n=pq which is the modulus used in the arithmetic calculations of the algorithm. The process of RSA algorithm is as follows.
1. Select p and q but both r prime numbers.
2. Calculate n= pq
3. Calculate z= (p-1)(q-1)
4. Select integer D which is relatively prime to 2. gcd ϕ (n)D= 1 (ϕ(n)= z)
5. Calculate ED= 1 mod (ϕ (n))

For encryption:

$$C = P^E \bmod n$$

Where P is plain text, C is cipher text(encryption)
For decryption (for calculating plain text)

$$P = C^D \bmod n$$

3.4 Blowfish encryption

Many of the encryption algorithms in today's time do not, show to public- most of them are protected by patents [1] (e.g. Khufu, REDOCII, and IDEA) in which secrecy provide by the governments. A blowfish algorithm is a symmetric block cipher which can take a key of variable length, from 32(4 Bytes) to 448bit (56 Bytes) [2] that makes it beneficial for exportable and domestic use. The elementary operator of Blowfish algorithm includes table lookup, addition and XOR [5].
Blowfish algorithm mainly contains two parts- the key expansion part and the data-encryption part [2]. Key expansion part changes a key length from 48 bits into 4168 bytes. It

contains P-array and having four S-boxes. P-array contains 18 of 32 bits sub keys, while each S-box contains 256 entries. Encryption of data performs by 16-round Fiestal structure.
The sub keys are calculated by using following steps-
1. First initialize P-array and S-boxes.
2. Using two XOR P-arrays of key bits 32 bits each.
3. Perform above methods for encrypting all zeros.
4. Obtain new output is P1 and P2.
5. Using sub keys encrypts new obtain output P1 and P2.
6. Then obtain new output is P3 and P4.
7. Repeat same steps upto 521 times in order to calculate new sub keys for P-array and the four S-boxes.

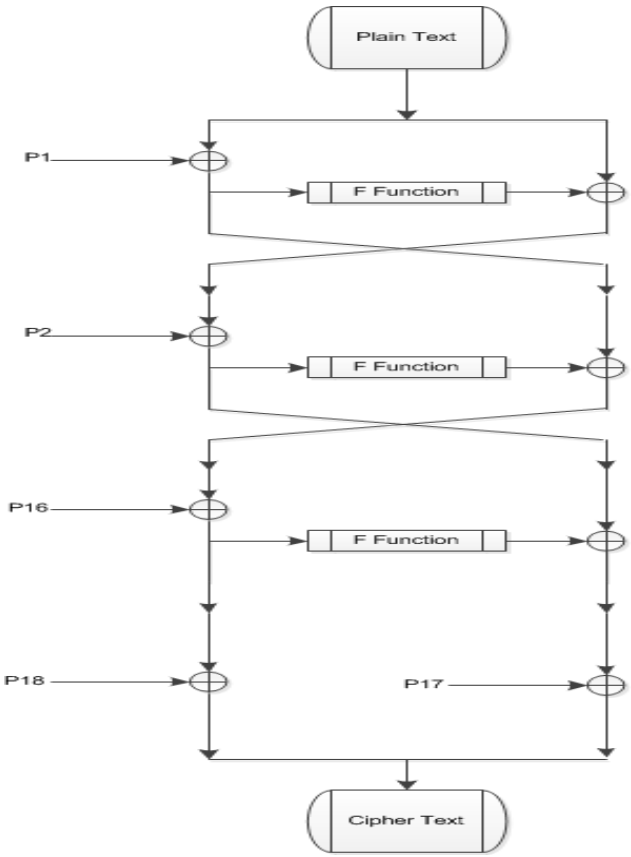

Fig4: each round action in Blowfish [8]

3.5 XOR encryption

In cryptography, a simple cipher is XOR cipher. Encryption algorithm can be operate on following principles-

$A \oplus 0 = A$,
$A \oplus A = 0$,
$(A \oplus B) \oplus C = A \oplus (B \oplus C)$,
$(B \oplus A) \oplus A = B \oplus 0 = B$

Where $\oplus$ is an exclusive disjunction (XOR). Sometimes it can be said that say that modulus 2 additions or subtraction. Using this logic text of a string can be encrypted using bitwise XOR operator to every character using a every key.
In this, encryption is done by-
1. First any plain text is input
2. Plain text is converted into ASCII representation.
3. after that converts into hexadecimal representation.
4. Converts it into binary equivalent representation.
5. Using XOR with key that converted into same plain text.
6. Cipher text is obtained.

Decryption is done by-
1. Input a cipher text
2. Perform XOR by using same key
3. Converting obtained binary code to hexadecimal code
4. After, converts it into ASCII code
5. Obtaining the plain text

4. RESULT ANALYSIS

Analyzing of different algorithms can performing by encryption and decryption on various size of data. An XOR algorithm converts the plain text into ASCII value and after that converts it into hexadecimal and then binary. In last, performing XORing with key which is easily performed. While in blowfish technique, first divide plain text into 64 bits blocks and then separated into left and right halves and performing iterative process using 8 to 448 with 16 Fiestal round with four S-boxes. Due to this it takes more space than XOR, DES and TDES.

Table1: space complexities of encryption algorithms of different size of data

Algorithm	before encryption	After encryption	After Decryption
XOR	160 KB	160 KB	160 KB
DES	160 KB	218 KB	160 KB
TDES	160 KB	390 KB	160 KB
BLOWFISH	160 KB	574 KB	160 KB

DES can also 16 rounds of Fiestal using 56 key with permutation which takes large space than XOR and TDES but less space than Blowfish which is variation of DES uses 168 key, that require large space than DES and XOR but smaller than Blowfish.
Assume that, encrypting same file by different encryption techniques then results are-

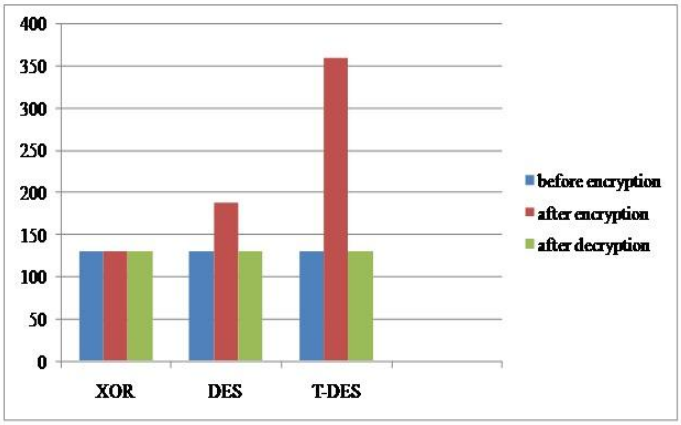

Fig5: Comparison encryption algorithms in terms of space

5. CONCLUSION AND FUTURE SCOPE

On the basis of implementation and their results, noticed that XOR is the fastest technique but it is very simple and acquire less space for storing intermediate cipher text which is approximately same as original plain text while other two i.e. DES and TDES are advanced technique then XOR, these are fast and secure due to its large size of the key length and having 16 Fiestal rounds with permutation in each round. While using Blowfish is faster than both DES and TDES because it uses four S-boxes in 16 Fiestel rounds bit it has more space complexity. After comparison of all algorithms find that TDES is more secure and technique can be chosen according to requirement. XOR algorithm is enhances as same as TDES because it is less secure and acquire less space than other algorithm. The space complexity is also compared with other algorithm such as RSA, DES, IDEA.

6. REFERENCES

[1] Suhaila Omer Sharif, S.P. Mansoor. 2010. Performance analysis of Stream and Block cipher algorithms. 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)

[2] Allam Mousa. 2005. Data Encryption Performance Based on Blowfish. 47th International symposium ELMAR, Zadar, Crotia

[3] P.Krishnamurthy. 2001. Encryption and Power Consumption in Wireless LANs-N. The Third IEEE Workshop on Wireless LANs - - Newton, Massachusetts

[4] Monika Agrawal, Pradeep Mishra. 2012. A Comparative Survey on Symmetric Key Encryption Techniques. International Journal on Computer Science and Engineering (IJCSE)

[5] Aamer Nadeem, dr M.Younus javed. 2005. A Performance Comparison of data Encryption Algorithms. Institute of Electrical and Electronics Engineers (IEEE)

[6] William Stallings. 2009. Cryptography and Network Security: Principles and practices. Dorling Kindersley (india) pvt ltd.

[7] Kruti R. Shah, Bhavika Gambhava. 2012. New Approach of Data Encryption Standard Algorithm. International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307. Volume-2, Issue-1

[8] Jawahar Thakur and Nagesh Kumar. 2011. DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis. International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459. Volume 1, Issue 2