ABSTRACT
With the fast progression of digital data exchange in electronic way, information security is becoming much more important in data storage and transmission. Cryptography has come up as a solution which plays a vital role in information security system against malicious attacks. The cryptography is most important aspect of communications security and becoming an important building block for computer security. This security mechanism uses some algorithms to scramble data into unreadable text which can be only being decoded or decrypted by party those possesses the associated key. These algorithms consume a significant amount of computing resources such as CPU time, memory and computation time. This paper analyses the performance of DES & 3DES which are widely used symmetric encryption algorithms i.e. Data Encryption Standard (DES) and triple Data Encryption Standard (3DES) in terms of time computation of encryption and decryption as well as avalanche effect of the both algorithms.

Keywords
Cryptography, DES, Triple DES, Avalanche effect

1. INTRODUCTION
Cryptography is about the design and analysis of mathematical techniques that enable secure communications in the presence of malicious adversaries. The security algorithms are classified into Symmetric Cipher Model and Asymmetric Cipher Model. As this paper presents the performance of DES & 3DES which are widely used symmetric encryption algorithms i.e. Data Encryption Standard (DES) and triple Data Encryption Standard (3DES) in terms of time computation of encryption and decryption as well as avalanche effect of the both algorithms.

2. DATA ENCRYPTION STANDARD
2.1 DES Encryption
The overall scheme for DES [1,2,3,4,16] encryption is illustrated in Figure 2.1. As with any encryption scheme, there are two inputs to the encryption function: the plaintext to be encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.

Looking at the left-hand side of the figure, we can see that the processing of the plaintext proceeds in three phases. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input. This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions. The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the pre output. Finally, the pre output is passed through a permutation that is the inverse of the initial permutation function, to produce the 64-bit cipher text. With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher.

The right-hand portion of Figure 3.1 shows the way in which the 56-bit key is used. Initially, the key is passed through a permutation function. Then, for each of the sixteen rounds, a subkey ($K_i$) is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each round, but a different sub key is produced because of the repeated shifts of the key bits.
2.2 DETAILS OF SINGLE ROUND:
Figure 2.2 shows the internal structure of a single round of DES. Again, begin by focusing on the left-hand side of the diagram. The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L (left) and R (right). As in any classic Feistel cipher, the overall processing at each round can be summarized in the following formulas:

\[ L_i = R_{i-1} \]
\[ R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \]

The round key \( K_i \) is 48 bits. The R input is 32 bits. This R input is first expanded to 48 bits by using a table that defines a permutation plus an expansion that involves duplication of 16 of the R bits (Table 3.2c). The resulting 48 bits are XORed with \( K_i \). This 48-bit result passes through a substitution function that produces a 32-bit output, which is permuted. The substitution consists of a set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.

![Figure 2.2: Internal Structure of Single round of DES](image)

2.3 DES Decryption
As with any Feistel cipher, decryption uses the same algorithm as encryption, except that application of the sub keys is reversed.

2.4 DES Encryption and Decryption Algorithms

**DES Encryption Algorithm**

function DES_Encrypt(M, K) where M = (L, R)

\[ M \leftarrow IP^{-1}(M) \]

for round 1 to 16

\[ K_i \leftarrow SK(K_{round}) \]

\[ L_i \leftarrow L \oplus F(R, K_i) \]

end

**DES Decryption Algorithm**

function DES_Decrypt(C, K) where C = (L, R)

\[ C \leftarrow IP(C) \]

for round 1 to 16

\[ K_i \leftarrow SK(K_{round}) \]

\[ L_i \leftarrow L \oplus F(R, K_i) \]

end

3. TRIPLE DES

3DES or the Triple Data Encryption Algorithm (TDEA)[18,19,21,22,23] was developed to address the obvious flaws in DES without designing a whole new cryptosystem. Data Encryption Standard (DES) uses a 56-bit key and is not deemed sufficient to encrypt sensitive data. 3-DES simply extends the key size of DES by applying the algorithm three times in succession with three different keys. The combined key size is thus 168 bits (3 times 56). TDEA involves using three 64-bit DEA keys (K1, K2, K3) in Encrypt-Decrypt-Encrypt (EDE) mode, that is, the plain text is encrypted with K1, then decrypted with K2, and then encrypted again with K3. Three-key 3DES has an effective key length of 168 bits and is defined as

\[ C = E(K3, D(K2, E(K1, P))) \]

The encryption and decryption process is shown in the figure 2.3

![Figure 2.3 The Encryption and Decryption process of 3DES](image)
3. Third option is, by using a key bundle of three identical keys (K1 = K2 = K3). This option is equivalent to DES Algorithm. In 3-DES the 3-times iteration is applied to increase the encryption level and average time. It is a known fact that 3DES is slower than other block cipher methods.

3.1 Triple DES Encryption and Decryption Algorithms

Triple DES Encryption Algorithm:
void DES_Encrypt(M,K1);
void DES_Decrypt(M,K2);
void DES_Encrypt(M,K3);

Triple DES Decryption Algorithm:
void DES_Decrypt(M,K1);
void DES_Encrypt(M,K2);
void DES_Decrypt(M,K3);

4. AVALANCHE EFFECT
A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the cipher text. In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the cipher text. This is referred to as the avalanche effect [12]. This paper presents the avalanche effect and positional based avalanche effect of DES and 3DES

5. EXPERIMENTAL RESULTS
The encryption and decryption time [6,7] of plain text for DES and 3DES is computed on Intel Core i5 2410M, 2.3GHz, 3GB RAM, Windows-7, 64 bit OS using Java and also observed the avalanche effect in both DES and 3DES

5.1 Time and Avalanche Effect
Comparison between DES and 3DES

<table>
<thead>
<tr>
<th>Parameter</th>
<th>DES</th>
<th>TRIPLE DES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time taken for encryption in milliseconds</td>
<td>92ms</td>
<td>176ms</td>
</tr>
<tr>
<td>Time taken for decryption process in milliseconds</td>
<td>10ms</td>
<td>29ms</td>
</tr>
<tr>
<td>Avalanche effect (Average)</td>
<td>30bits</td>
<td>32bits</td>
</tr>
<tr>
<td>Position based avalanche effect (Average)</td>
<td>32bits</td>
<td>33bits</td>
</tr>
</tbody>
</table>

5.2 Avalanche Effect

<table>
<thead>
<tr>
<th>No. of input bits changed in Plain Text</th>
<th>DES</th>
<th>3DES</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of bits changed in Cipher Text</td>
<td>31</td>
<td>34</td>
</tr>
<tr>
<td>No. of bits changed in Plain Text</td>
<td>36</td>
<td>25</td>
</tr>
<tr>
<td>No. of bits changed in Cipher Text</td>
<td>36</td>
<td>35</td>
</tr>
</tbody>
</table>

5.3 Graphical Representation of Avalanche Effect

5.4 Position based Avalanche Effect

<table>
<thead>
<tr>
<th>Change in Input</th>
<th>No. Of Outputs changed</th>
<th>bits changed</th>
<th>No. of output bits changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES</td>
<td>3DES</td>
<td>DES</td>
<td>3DES</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
<td>34</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>31</td>
<td>30</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
<td>30</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>31</td>
<td>30</td>
<td>20</td>
</tr>
</tbody>
</table>
5.5 Graphical Representation of Position based Avalanche Effect

6. CONCLUSION
According to research done and literature survey on DES and 3DES it can be found that 3DES algorithm is most efficient in terms of avalanche effect and also this paper presents the encryption time is more than the decryption time for both algorithms.

7. REFERENCES


[23] Network security (second edition) by Kaufman-Perlman-Speciner