A Novel Approach for Data Encryption Standard Algorithm


Abstract—Now a day's providing Security for data is complicated task we have so many security methods that are implemented and deployed but out of them few are using and serving the needs of society. And we can't say that any algorithm is perfect and avoids threats. The main goal of any design of any encryption algorithm must be security against unauthorized attacks. Within the last decade, data in both the private and public sectors are increased which requires Availability, Authentication, Confidentiality, Integrity. In this paper we are considering The DES algorithm that defines the mathematical steps that transform original text (plain text) into a cipher text (secret code) and also transform the cipher text back to the original text. Here introduces a new method to enhance the performance of the Data Encryption Standard (DES) algorithm. This is done by replacing the 8/32 S-Box instead of 6/4 S-Box. The output of each S-Box undergoes AND and XOR operation before going to the permutation P. In this paper we also proposed a new operation Addition modulo instead predefined XOR operation applied during the 16 round of the standard algorithm.

Index Terms— DES, Encryption, Decryption, asymmetric cryptography, symmetric cryptography.

I. INTRODUCTION

Cryptography: is usually referred to as the study of secret. It is probably most important aspect of communication security and is becoming increasingly important as a basic building block for computer security[3]. Data that can be read and understood without any special measures is called plaintext. The method of representing plaintext in such a way as to hide its substance is called encryption. Encrypting plaintext results in unreadable form is called cipher-text. The process of reverting cipher-text to its original plaintext is called decryption. There are two techniques used for data encryption and decryption, which are: Asymmetric Cryptography in this sender and recipient use different keys then it is known as asymmetrical or public key cryptography. The key used for encryption is called the public key and the key used for decryption is called the private key.

Symmetric Cryptography If sender and recipient use the same key then it is known as symmetrical or private key cryptography as shown in Fig 1. It is always suitable for long data streams. One of the symmetric block encryption algorithm is DES[9].

II. DATA ENCRYPTION STANDARD

The DES (Data Encryption Standard) algorithm is the most widely used encryption algorithm in the world [15]. For many years, and among many people, “secret code making” and DES have been synonymous. The DES was published by the United States’ National Bureau of Standards in January 1977 as an algorithm to be used for unclassified data (information not concerned with national security)[1]. The Data Encryption Standard (DES), as specified in FIPS Publication 46-3, is a block cipher operating on 64-bit data blocks as shown in Fig 2. The encryption transformation depends on a 56-bit secret key and consists of sixteen Feistel iterations surrounded by two permutation layers: an initial bit permutation IP at the input, and its inverse IP−1 at the output. The structure of the cipher is depicted in Figure 2. The decryption process is the same as the encryption, except for the order of the round keys used in the Feistel iterations.[14] The 16-round Feistel network, which constitutes the cryptographic core of DES, splits the 64- bit data blocks into two 32-bit words, LBlock and RBlock (denoted by L0 and R0). In each iteration (or round), the second word R'i is fed to a function f and the result is added to the first word L'i. Then both words are swapped and the algorithm proceeds to the next iteration.

Manuscript received June 10, 2013.

A Novel Approach for Data Encryption Standard Algorithm

The function f of DES algorithm is key dependent and consists of 4 stages.

1. Expansion (E): The 32-bit input word is first expanded to 48 bits by duplicating and reordering half of the bits.\[7\]
2. Key mixing: The expanded word is XORed with a round key constructed by selecting 48 bits from the 56-bit secret key, a different selection is used in each round.
3. Substitution. The 48-bit result is split into eight 6-bit words which are substituted in eight parallel 6×4-bit S-boxes. All eight S-boxes are different but have the same special structure.
4. Permutation (P): The resulting 32 bits are reordered according to a fixed permutation before being sent to the output. The modified R Block is then XORed with LBlock and the resultant fed to the next RBlock register. The unmodified R Block is fed to the next L Block register. With another 56 bit derivative of the 64 bit key, the same process is repeated.

Pseudo Code: Data Encryption Standard:

INPUT: plaintext p1 . . . p64; 64-bit key K=k1 . . . k64 (includes 8 parity bits).
OUTPUT: 64-bit cipher text block C=c1 . . .c64.

1. (key schedule) Compute sixteen 48-bit round keys Ki, from K.
   Note: Where k=64 bits out of which 8 parity bits are discarded outcome is 56 bits, after Left circular shift and PC2 which results 48 bit key.

2. (L0, R0) ← IP(p1, p2,...p64) (Use IP Table to permute bits; split the result into left and right 32-bit halves L0=p58p50 . . . p8,R0=p57p49 . . . p7)

3. (16 rounds) for i from 1 to 16, compute Li and Ri as follows:
   3.1. Li=Ri-1
   3.2. Ri = Li-1 XOR f (Ri-1, Ki)

   where f(Ri-1, Ki) = P(S(E(Ri-1) XOR Ki)), computed as follows:
   (a) Expand Ri-1 = r32r1r2 . . . r32 r1 from 32 to 48 bits, M ← E(Ri-1).
   (b) M’ ← M XOR Ki. Represent M’ as eight 6-bit character strings: M’= (B1 . . . B8)
   (c) M” ← S1(B1), S2(B2),..., S8(B8)). Here Si(Bi) maps the 4-bit entry in row r and column c of Si
   (d) M”’ ← P (M’). (Use P per table to permute the 32 bits of M”=m1m2 . . . m32, yielding m16m7 . . . m25)

4. b1b2 . . . b64 ← (R16, L16). (Exchange final blocks L16, R16.)
5. C ← IP-1 (b1b2 . . . b64).
6. End.

III. PROPOSED DES ALGORITHM

In this paper we proposed a new improvement to the DES algorithm which makes the use of the new operation known as addition modulo (+). it takes two inputs and performs addition and resulting output assume like x. later perform x mod 2^w Where w is the number of bits that depends on given input.

Example: x and y are the Inputs
X=1100 1000
Y=1000 1111
X’ is obtained by performing x+y
X’= 1 0101 0111

Carry can be thrown off (or) perform modulo 2^8
X’ is converted to decimal number
X’= 343 mod 2^8 = 87
Binary equivalent of x’ is 0101 0111
To find original x value perform following operation
X=x’+(−y)
To obtain (−y) = 2^8y =>256-143=113
Perform X’+(−y) which results
Original x
0111 0001
0101 0111

1100 1000 ← original x value

The graphical representation of the proposed Des as shown in Fig.3

Fig. 3 Proposed DES Algorithm.

A. Algorithm of modified data encryption standard with addition modulo operation

INPUT: plaintext p1 . . . p64; 64-bit key K=k1 . . . k64 (includes 8 parity bits).
OUTPUT: 64-bit cipher text block C=c1 . . .c64.

1. (key schedule) Compute sixteen 48-bit round keys Ki, from K.
Note: Where k=64 bits out of which 8 parity bits are discarded outcome is 56 bits, after Left circular shift and PC which results 48 bit key.

2. \((L0, R0) \leftarrow \text{IP}(p1, p2, \ldots p64)\) (Use IP Table to permute bits; split the result into left and right 32-bit halves \(L0=p58p50 \ldots p8, R0=p57p49 \ldots p7\))
3. (16 rounds) for \(i\) from 1 to 16, compute \(Li\) and \(Ri\) as follows:
   3.1. \(Li=Ri-1\)
   3.2. \(Ri = Li-1 \text{addition mod} 2^{32} f(Ri-1, Ki)\) where \(f(Ri-1, Ki) = P(S(E(Ri-1) \text{XOR} Ki))\), computed as follows:
      (a) Expand \(Ri-1 = r32r1r2 \ldots r32 r1\) from 32 to 48 bits, \(M \leftarrow E(Ri-1)\).
      (b) \(M' \leftarrow M \text{ XOR} Ki\). Represent \(M'\) as eight 6-bit character strings: \(M' = (B1 \ldots B8)\).
      (c) \(M'' \leftarrow F'\) where function \(F' = (((s1 \text{AND} s2) \text{XOR} s3) \text{AND} s4)\text{XOR} s5) \text{AND} s6\). Here \(s(i)\) maps to the 8/32 S-Box that consist of 256 entries.
      (d) \(M''' \leftarrow P(M'')\). (Use \(P\) per table to permute the 32 bits of \(M'' = m1m2 \ldots m32\), yielding \(m16m7 \ldots m25\))
   4. \(b1b2 \ldots b64 \leftarrow (R16, L16)\). (Exchange final blocks \(L16, R16\).)
   5. \(C \leftarrow \text{IP}^{-1}(b1b2 \ldots b64)\).
   6. End.

During each round \(i\), the right half of the block is expanded to 48 bits and XORed with a 48 bit internal key \(Ki\) derived from \(K\). the result then passes through 6 s-boxes which are nonlinear substitutions results 32 output bits from 8 input bits.

The 32 bit result undergoes AND, XOR operations and is then permuted and performs addition modulo232 with left half of the block. As shown in Fig.4 Finally the two halves of the block are swapped before going through the next round. After completing 16 rounds the result will be undergone for \(IP^{-1}\) and finally we will get cipher text of 64 bits length. For the input given above the following TABLE.I shows 16 rounds and resulting output.

**TABLE.I: Result of proposed DES 16 Rounds**

<table>
<thead>
<tr>
<th>Round</th>
<th>Sub Key(Ki)</th>
<th>Left Bits(Li)</th>
<th>Right Bits(Ri)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>CC00CCFF</td>
<td>F0AFA0AA</td>
<td>CE620D18</td>
</tr>
<tr>
<td>1</td>
<td>1B02EFFC7072</td>
<td>F0AFA0AA</td>
<td>CE620D18</td>
</tr>
<tr>
<td>2</td>
<td>79AEB9DBC9EF</td>
<td>CE620D18</td>
<td>22D1F6C2</td>
</tr>
<tr>
<td>3</td>
<td>55FC8A42699</td>
<td>22D1F6C2</td>
<td>D684E29</td>
</tr>
<tr>
<td>4</td>
<td>72ADD6DB351D</td>
<td>DE6B4E29</td>
<td>AB5BBF24</td>
</tr>
<tr>
<td>5</td>
<td>7CEC07EB53AE</td>
<td>AB5BBF24</td>
<td>E4977040</td>
</tr>
<tr>
<td>6</td>
<td>63A53E5072BF</td>
<td>E4977040</td>
<td>2BB88B30</td>
</tr>
<tr>
<td>7</td>
<td>EC84B7F618BC</td>
<td>2BB88B30</td>
<td>2531F060</td>
</tr>
<tr>
<td>8</td>
<td>F78A3AC13BF B</td>
<td>2531F060</td>
<td>D0B8BCB2</td>
</tr>
<tr>
<td>9</td>
<td>E0DBEDE781</td>
<td>D0B8BCB2</td>
<td>7436155C</td>
</tr>
<tr>
<td>10</td>
<td>B1F347BA446F</td>
<td>7436155C</td>
<td>56FFCE0E</td>
</tr>
<tr>
<td>11</td>
<td>215FD33ED386</td>
<td>56FFCE0E</td>
<td>D66AAA00</td>
</tr>
<tr>
<td>12</td>
<td>7571F5946E9</td>
<td>D66AAA00</td>
<td>587FC4EA</td>
</tr>
<tr>
<td>13</td>
<td>97C4D1FABA41</td>
<td>587FC4EA</td>
<td>A6FFD004</td>
</tr>
<tr>
<td>14</td>
<td>5F43B7F2E73A</td>
<td>A6FFD004</td>
<td>FAA3CA30</td>
</tr>
<tr>
<td>15</td>
<td>BF918D33F0A</td>
<td>FAA3CA30</td>
<td>E9FF05C</td>
</tr>
<tr>
<td>16</td>
<td>EB3D8B0E1F5S</td>
<td>E9FF05C</td>
<td>5BC44D34</td>
</tr>
</tbody>
</table>

**B. Avalanche Effect**

A change in one bit of the plain text 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.
The following example shows the result when the first bit of the plain text is changed that results more changes in the resulting bits of the cipher text.

Plain text: 1123456789ABCDEF

Altered Bit

<table>
<thead>
<tr>
<th>Round</th>
<th>Input</th>
<th>Left Bits (Li)</th>
<th>Right Bits (Ri)</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1stRound</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0123456789A</td>
<td>BCDEF11234</td>
<td>56789ABCD</td>
<td>EF</td>
<td>CC00CCFF</td>
</tr>
<tr>
<td>1</td>
<td>CC00CCFF</td>
<td>0AAF0AACC</td>
<td>01CCFFFA1A</td>
<td>AF0AA</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

As we towards a society where automated information resources are increased and cryptography will continue to increase in importance as a security mechanism. DES is now considered to be insecure for some applications like banking system. There are also some analytical results which demonstrate theoretical weaknesses in the cipher. So it becomes very important to augment this algorithm by adding new levels of security to make it applicable. By adding modified S-Box design, modifies function implementation and replacing the old XOR by a new operation as proposed by this thesis to give more robustness to DES algorithm and make it stronger against any kind of intruding. This new algorithm gives avalanche effect than the original DES algorithm and also solves cryptanalysis attack.

ACKNOWLEDGMENT

We take this opportunity to acknowledge those who have been great support and inspiration through the research work. Our sincere thanks to Mrs. B.Renuka Devi, head of the department of CSE to her diligence and motivation. Special thanks to Vignan’s Lara Institute of Science and Technology, for giving us such a nice opportunity to work in the great environment and for providing the necessary facilities during the research and encouragement from time to time. Thanks to our colleagues who have been a source of inspiration and motivation that helped to us and to all other people who directly or indirectly supported and help us to fulfill our task. Finally, we heartily appreciate our family members for their motivation, love and support in our goal.

REFERENCES

[7] Network security and cryptography by bernard menezes

Prashanti.G is Assistant Professor at Department of Computer Science and Engineering, Vignan’s Lara Institute of Technology and Science, Vadlamudi, Guntur, Andhra Pradesh, India. She has received her M.Tech degree from Vignan Engineering College, Vadlamudi., Guntur, Andhra Pradesh, India

Deepti.S is Assistant Professor at Department of Computer Science and Engineering, Vignan’s Lara Institute of Technology and Science, Vadlamudi, Guntur, Andhra Pradesh, India. She has received her B.Tech degree from K.L College of Engineering, Guntur, Andhra Pradesh, India in 2008. She has received her M.Tech degree from Nalanda Institute of Engineering and Technology, Guntur, Andhra Pradesh, India

Sandhya Rani.K is Assistant Professor at Department of Computer Science and Engineering, Vignan’s Lara Institute of Technology and Science, Vadlamudi, Guntur, Andhra Pradesh, India. She has received her M.Tech degree from Vignan Engineering College, Vadlamudi., Guntur, Andhra Pradesh, India