World of Computer Science and Information Technology Journal (WCSIT)
ISSN: 2221-0741
Vol. 1, No. 3, 88-91, 2011

A proposed Modified Data Encryption Standard algorithm by Using Fusing Data Technique

Alaa H. AL-Hamami
Computer Sciences Dept.
Amman Arab University
Amman, Jordan
Alaa_hamami@yahoo.com

Mohammad A. AL-Hamami
Computer Sciences Dept.
Delmon University
Manama, Bahrain
mohammad.alhamami@yahoo.com

Soukaena H. Hashem
Computer Sciences Dept.
University of Technology
Baghdad, Iraq
soukaena_hassan@yahoo.com

Abstract— Data Encryption Standard (DES) is a block cipher that encrypts data in 64-bit blocks. A 64-bit block of plaintext goes in one end of the algorithm and a 64-bit block of cipher text comes out of the other end. Blowfish is a block cipher that encrypts data in 8-byte blocks .Blowfish consists of two parts: a key-expansion part and a data-encryption part. Key expansion converts a variable-length key of at most 56 bytes (448 bits) into several subkey arrays totaling 4168 bytes. Blowfish has 16 rounds, such as DES. In this research the fusion philosophy will be used to fuse DES's with blowfish and Genetic Algorithms by taking the strong points in all of these techniques to create a proposed Fused DES-Blowfish algorithm. The proposed algorithm is presented as a modified DES depending on the advantage in key generation complexity in blowfish and advantage of optimization in Genetic Algorithm to give the optimal solution. The solution will be the depended tool for creation of the strong keys.

Keywords- Fusing; Blowfish; Genetic Algorithm; Strong keys; and Data Encryption Standard.

I. INTRODUCTION

Despite its popularity, DES has been plagued with controversy. Some cryptographers objected to the closed-door design process of the algorithm. The debate about whether DES's key is too short for acceptable commercial security has raged for many years, but recent advances in distributed key search techniques have left no doubt in anyone's mind that its key is simply too short for today's security applications. Blowfish is a Festal network consisting of 16 rounds (see Figure 1). The input is a 64-bit data element, x. Divide x into two 32-bit halves: xL, xR. Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order. Implementations of Blowfish that require the fastest speeds should unroll the loop and ensure that all subkeys are stored in cache [1-8].

II. PROPOSED MODIFIED DES ALGORITHM

This research aims to fuse DES algorithm with Blowfish algorithm and Genetic Algorithm (GA). The suggested fusion is in key generation. To explain the proposed system in details, we suggest fused DES-Blowfish algorithm with the following features. The suggested fused DES-Blowfish must have specific characteristics; these are: Security must be completely specified, easy to understand, public, available to all users, efficient to use, able to be validated, and exportable.

A. *Description of the modified DES Algorithm*

The modified DES algorithm is a block cipher; it encrypts data in 64-bit blocks. A 64-bit block of plaintext goes in one end of the algorithm and a 64-bit block of ciphertext comes out the other end. Fused DES-Blow (modified DES) is a symmetric algorithm: This algorithm has two keys which are used for both encryption and decryption (except for minor differences in the key schedule). We suggest the proposed fused algorithm to have two keys, one in the left side called Left Key and another on the right side called Right Key.

Left Key has initial key with length of 768 bits and 16 block of 48-bits which is called Pi from both of them the 16th left sub-keys are generated as in the following steps: The 16th left sub-keys are calculated as follows:

1. Initialize the 768 and 16 P-array each array have 48 bit. Both of them have initial keys. The 768 bits and the 16th array will be taken randomly as hexadecimal digits.
   Initial key = 0x768adfc……
   P1 = 0x243f6a887321
   P2 = 0x85a308d3cd89

P3 = 0x13198a2e3562
P4 = 0x0370734fdca2
…………..
P16=0x6abcf3429821

2. Converting the hexadecimal to a binary.
3. XOR the first 48-bit of initial key with the first array P1 to create the first 48-bit left sub-key which supports the first round from the left.
4. Continue XORing the second 48-bit of initial key with the second array P2 to create the second 48-bit left sub-key which supports the second round from the left.
5. Until XORing the last 48-bit of initial key with the last array P16 to create the 16th 48-bit left sub-key which support the 16th round from the left.

Right Key has initial key 100 of 48-bits taken as initial generation for genetic algorithm. By applying the proposed GA to get best 16 of 48-bit to be the right sub-keys. The basic parameters for this proposed GA are:

1. 100 seeds that each seed has number of bits equal to 48 bit.
2. Here the proposed evaluation function for each key is *the hamming distance function* that compares the keys with known weak keys.
3. *Two-point crossover* is the most suitable crossover operator, where a crossover points on the genetic code which is selected randomly, and two parent frames are interchanged at this point.
4. A mutation operator can prevent any single bit from converging to a value through the entire population and, more important, it can prevent the population from converging and stagnating at any local optima.
5. Population size, *pop-size* = 100 (the parameter was already used), Probability of crossover, PC = 1, Probability of mutation, PM = 0.001 (the parameter will be used in a mutation operation).
6. Continue with genetic processing until obtain the optimized key to be the master key.

Both keys can be changed at any time. All security rests within the key. At its simplest level, the algorithm is nothing more than a combination of the two basic techniques of encryption: confusion and diffusion. The fundamental building block of Fused DES-Blow is a single combination of these techniques (a substitution followed by a permutation) on the text, based on the ***two keys***. This is known as a round. Fused DES-Blow has 16 rounds; it applies the same combination of techniques on the plaintext block 16 times see “Fig. 1”.

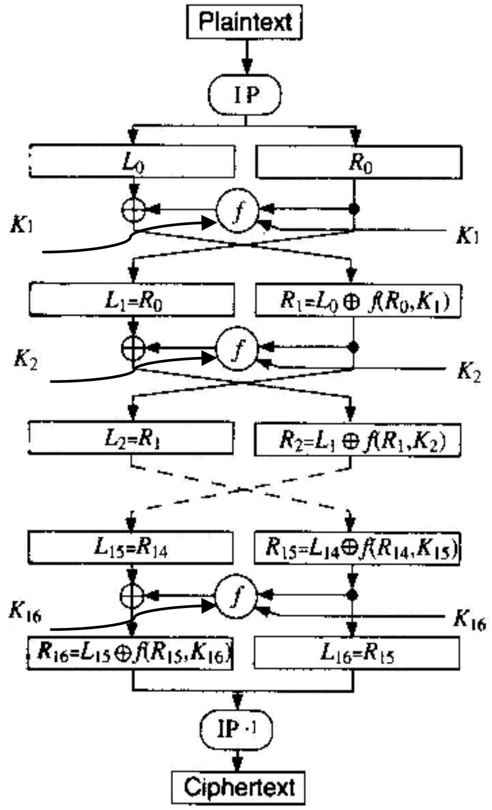


Figure 1. Fused DES- Blow

B. Outline of the Algorithm

The basic process in enciphering a 64-bit data block using the Fused DES-Blow consists of:

- An initial permutation (IP)
- 16 rounds of a complex ***two keys*** depredating on calculation of f.
- Final permutation, being the inverse of IP

In each round see “Fig. 2”, take the sequenced 48 key bits then XORing it with the corresponding 48 of left and right sub-keys.

The right half of the data is expanded to 48 bits via an expansion permutation, combined with 48 bits of right sub-key via an XOR, then again combined with 48 bits of left sub-key. Finally sent through 8 S-boxes producing 32 new bits, and permuted again. These five operations make up Function f.

The output of Function f is then combined with the left half via another XOR. The result of these operations becomes the new right half; the old right half becomes the new left half. If Bi is the result of the ith iteration, Li and Ri are the left and right halves of Bi, Ki is the 48-bit key for round i, and f is the function that does all the substituting and permuting and XORing with the key, then a round looks like:

$Li = R_{j-1}$

$$Ri = (L_{i-1} \text{ Xor } f\,(R_{i-1}, Kl_i\,, Kr_{i,}\ )\ )$$

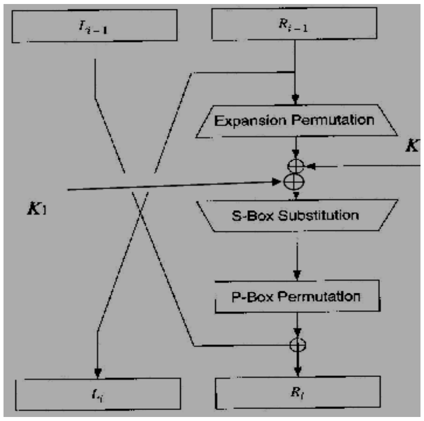


Figure 2. One round of DES.

C. Decrypting Fused DES-Blowfish

After all the substitutions, permutations, XORs, and shifting around, you might think that the decryption algorithm is completely different and just as confusing as the encryption algorithm. On the contrary, the various operations were chosen to produce a very useful property: The same algorithm works for both encryption and decryption. With DES it is possible to use the same function to encrypt or decrypt a block. The only difference is that the keys must be used in the reverse order. That is, if the encryption keys for each round are K1, K2, K3, . . . , K16, for right then K1, K2, K3, . . . , K16, for left, so the decryption keys are K16, K15, K14, . . . , K1, for left then K16, K15, K14, . . . , K1, for right.

III. IMPLEMENTATION

The implementation of the proposed fused DES-Blow was done under visual basic programming language. To explain the proposed algorithm, “Fig. 3” will explain the main window which displays the basic four sequenced steps of the proposed fused DES-Blow.

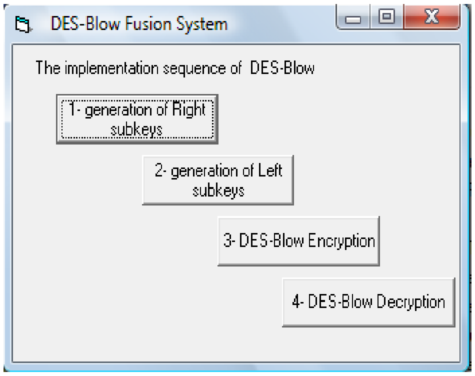
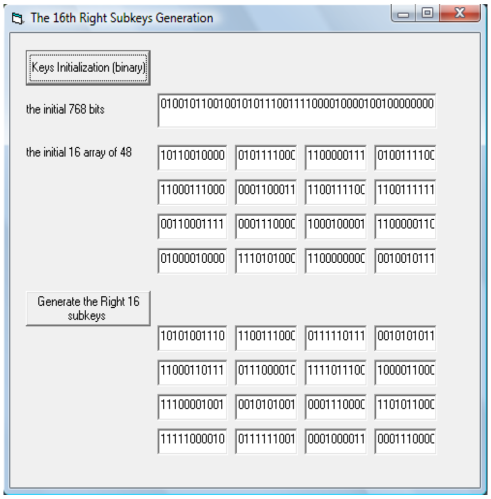

DES-Blow Fusion System
The implementation sequence of DES-Blow
1- generation of Right subkeys
2- generation of Left subkeys
3- DES-Blow Encryption
4- DES-Blow Decryption

Figure 3. The main window of DES-Blow system.

See “Fig. 4” explains the right 16 subkeys generation and “Fig. 5” explains the left 16 subkeys generation.


The 16th Right Subkeys Generation
Keys Initialization (binary)
Generate the Right 16 subkeys

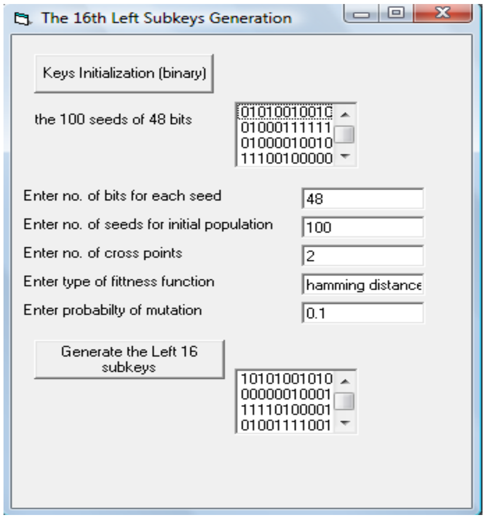
the initial 768 bits 01001011001001010111001111000010000100100000000

the initial 16 array of 48

```
10110010000  0101111000  1100000111  0100111100
11000111000  0001100011  1100111100  1100111111
00110001111  0001110000  1000100001  1100000110
01000010000  1110101000  1100000000  0010010111
```

```
10101001110  1100111000  0111110111  0010101011
11000110111  0111000010  1111011100  1000011000
11100001001  0010101001  0001110000  1101011000
11111000010  0111111001  0001000011  0001110000
```

Figure 4. Right 16 subkeys generation.


The 16th Left Subkeys Generation
Keys Initialization (binary)
Generate the Left 16 subkeys

the 100 seeds of 48 bits

```
01010010010
01000111111
01000010010
11100100000
```

Enter no. of bits for each seed 48

Enter no. of seeds for initial population 100

Enter no. of cross points 2

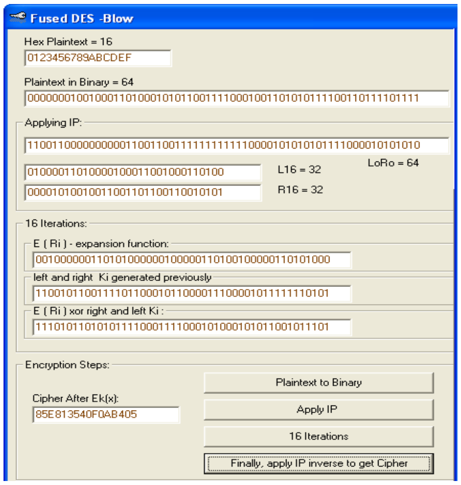
Enter type of fittness function hamming distance

Enter probabilty of mutation 0.1

```
10101001010
00000010001
11110100001
01001111001
```

Figure 5.Left 16 subkeys generation.

See “Fig. 6” explains the encryption process and “Fig. 7” explains the decryption process.



Hex Plaintext = 16
0123456789ABCDEF

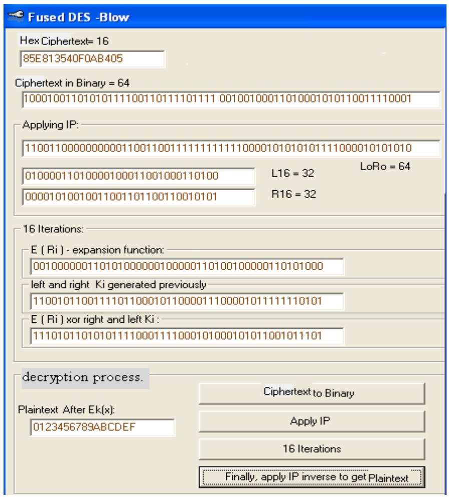
Plaintext in Binary = 64
0000000100100011010001010110011110001001101010111100110111101111

Applying IP:
1100110000000000110011001111111111110000101010101111000010101010
01000011010000100011001000110100 L16 = 32
00001010010011001101100110010101 R16 = 32
LoRo = 64

16 Iterations:
E (Ri) - expansion function:
001000000110101000000010000001101001000000110101000
left and right Ki generated previously
110010110011110110001011000011100001011111110101
E (Ri) xor right and left Ki :
111010110101011110001111000101000101011001011101

Encryption Steps:
Cipher After Ek(x):
85E813540F0AB405

Figure 6. Encryption process.



Hex Ciphertext= 16
85E813540F0AB405

Ciphertext in Binary = 64
10001001101010111100110111101111 00100100011010001010110011110001

Applying IP:
1100110000000000110011001111111111110000101010101111000010101010
01000011010000100011001000110100 L16 = 32
00001010010011001101100110010101 R16 = 32
LoRo = 64

16 Iterations:
E (Ri) - expansion function:
001000000110101000000010000001101001000000110101000
left and right Ki generated previously
110010110011110110001011000011100001011111110101
E (Ri) xor right and left Ki :
111010110101011110001111000101000101011001011101

decryption process.
Plaintext After Ek(x):
0123456789ABCDEF

Figure 7. Decryption process

IV. CONCLUSIONS

Using blowfish key generation method for generating all right 48-bits sub-keys to increase the key complexity since all these sub-keys generated from initial 768-bit key XORing with the P arrays (each 48-bits). Using GA as a key generation to provide the left 48-bit sub-keys increase the reliability of the keys as a secure keys since it depend on hamming distance function for fitness to avoid all types of week keys. Encryption with two keys instead of one key already will increase the efficiency of cryptography since it increases the length of the key so the attacker must try 2^2n if the length of stand key was n.

V. DISCUSSION AND RESULTS

Previous algorithms DES and blowfish each of them have only one initial key n-bits derived from it all subkeys, so by brute force attack on initial key (2^n) all subkeys will be computed very easy. The strong point in our proposed fused DES-Blow algorithm is having two initial keys n-bits and m-bits. The generation of the subkeys for each initial key depends on different approach. The left subkeys depend on blowfish key generation and the right subkeys depend on genetic algorithm. So the block of plaintext in each round will be encrypted twice by two different subkeys. So the cryptanalysis computation will be the double in the proposed algorithm, since he/she must try to find two initial keys instead of one and generate 32 subkey instead of 16. Table 1 shows the comparisons results we reached to after implementing the three algorithms in the same environment.

Table 1. Comparisons Results

Metrics	DES	Blowfish	Fused DES-Blow
Random signature	70%	78%	85%
Random subkeys	50%	70%	90%
Optimality	84%	84%	92%
Computation Speed	80%	77%	75%
Cryptanalysis immune	40%	50%	80%

REFERRENCES

[1]. W. Stalling, “Network security essential: application and standard”, William stalling books for network and data communication technology, 2001.

[2]. B. Beckett, “Introduction to cryptography and PC security”, McGraw-Hill companies, 1997.

[3]. M. A. Al-hamami, S. H. Hashem, “Improving performance and random signature schemes in twofish cryptosystem”, journal of Al_Rafidian, 2006.

[4]. A. H. Al-hamami, M. A. Al-hamami and S. H. Hashem, "A Proposed Modifications to Improve the Performance of Blowfish Cryptography Algorithm", First National Information Technology Symposium (NITS 2006) Bridging the Digital Divide: Challenge and Solutions, King Saud University, Riyadh, Kingdom of Saudi Arabia, 5-7 Feb. 2006.

[5]. V. Chatzigiannakis, S. Papavassiliou, G. Androulidakis and B. Maglaris, “On the realization of a generalized data fusion and network anomaly detection framework”, Fifth International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP’06), Patra, Greece, July 2006.

[6]. Z. Wang, D. Ziou, C. Armenakis, D. Li, and Q. Li, “A comparative analysis of image fusion methods,” IEEE Trans. Geosci. Remote Sensing, vol. 43, no. 6, pp. 1391–1402, June 2005.

[7]. B. Aiazzi, L. Alparone, S. Baronti, and A. Garzelli, “Context-driven fusion of high spatial and spectral resolution data based on oversampled multiresolution analysis,” IEEE Trans. Geosci. Remote Sensing, vol. 40, no. 10, pp. 2300–2312, Oct. 2002.

[8]. F. Laporterie-D´ejean, H. de Boissezon, G. Flouzat, and M.-J. Lef`evre-Fonollosa, “Thematic and statistical evaluations of five panchromatic/multispectral fusion methods on simulated PLEIADES-HR images,” Inform. Fusion, vol. 6, no. 3, pp. 193–212,Sep. 2005.