Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 66 (2014) 20924-20926

ISSN: 2229-712X

Data Security with Hybrid Aes-Des

Minal Moharir[1,*] and A V Suresh[2]

[1]Department of Information Science and Engineering, R V College of Engineering, Bangalore, India.
[2]Dean Academics, R V College of Engineering, Bangalore, India.

ARTICLE INFO

Article history:
Received: 5 June 2013;
Received in revised form:
20 January 2014;
Accepted: 23 January 2014;

Keywords
Data Security,
Symmetric Key Cryptography,
Aes,
Des.

ABSTRACT

The problem with AES, most extensively used encryption, is that it uses many multivariate equations which are linear in nature . Thus it can be broken using algebraic cryptanalysis. This provides a serious threat as AES was considered to be unbreakable and thus it was used in many encryption systems. The DES predecessor of AES has problem of multivariate linear equation and speed. By exploiting the key steps and vulnerabilities in these algorithms it is possible to break through AES and DES secured data.

Introduction

The problem with AES[1], most extensively used encryption, is that it uses many multivariate equations which are linear in nature [3]. Thus it can be broken using algebraic cryptanalysis. This provides a serious threat as AES was considered to be unbreakable and thus it was used in many encryption systems. The DES predecessor of AES has problem of multivariate linear equation and speed. By exploiting the key steps and vulnerabilities in these algorithms it is possible to break through AES and DES secured data [4].

Therefore the paper proposed a hybrid AES-DES algorithm. This improved algorithm contains the features of AES along with DES algorithm. The use of AES inside a DES[2] structure highly increases the number of permutation combinations used in the structure. A number of keys required for both the algorithms further increase the encryption strength of the algorithm.

This paper presents a new hybrid design 128 bit key AES-DES algorithm, as security enrichment. Standard AES-128 is implemented using 10-rounds, standard DES is implemented using 18-rounds. A proposed hybrid algorithm is implemented in 10-rounds.

Instead of implementing AES transformation in intermediate rounds [3-7]. The first & last round is from AES cipher and the intermediate 8-rounds are implemented using DES's feistel structure. The implementation details are as given in section3.

Related Work

According literature [4] has implemented hybrid AES-DES image encryption. The other few designs are proposed [4-5] which combines both the algorithm together. It carries the basic flaws of both the algorithms together. The literature [5] had proposed hybrid AES-DES. In this paper the authors had implemented AES inside the DES feistel structure. This makes the design more complicated & time consuming.

Algorithm Implementation

As described in [1], each AES data block consists of 128 bits (16 bytes) of data. During the encryption and decryption processes these 16 bytes will create a changeable (4*4) array called the state array. This state array is processed by first round which is substitution step in AES. The complete architectural overview of propose algorithm is as shown in Figure 1.

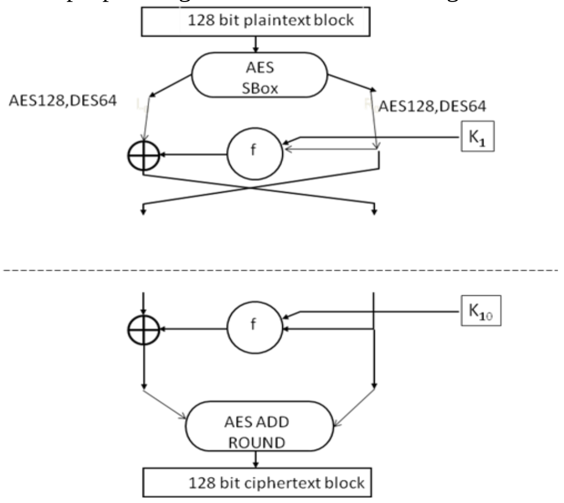


Figure 1. Hybrid AES-DES

First Step AES SubBytes Transformation

The first step in proposed algorithm is AES SubBytes transformation. The SubBytes transformation uses AES substitution table(S-box) to provide non-linear substitution. To meet the non-linearity requirement, the S-box is constructed by combining two transformations which include an inverse function and an invertible affine transformation. In the first transformation, the elements of GF(2^8) are represented as polynomials which have degrees less than eight, with coefficients in GF(2). Multiplications are carried out modulo the irreducible polynomial x8 + x4 +x3 + x + 1, and the multiplicative inverses are defined accordingly. The element {00} is mapped to itself. As the first transformation has a simple algebraic expression, which may lead to attacks such as algebraic attacks, it is essential to apply second transformation is an affine transformation (over GF (2)).

The substitution table is as per standard AES S-Box, with all elements in hexadecimal; form. For example, if the input to SubBytes transformation is {6f}, the output would be {a8}.

Tele:
E-mail addresses: moharirminal@gmail.com

Intermediate 8-rounds: DES Fiestiel Cipher

In cryptography, a Feistel cipher is a symmetric structure used in the construction of block ciphers, named after the German-born physicist and cryptographer Horst Feistel who did pioneering research while working for IBM (USA); it is also commonly known as a Feistel network [51]. A large proportion of block ciphers use the scheme, including the Data Encryption Standard (DES). The Feistel structure has the advantage that encryption and decryption operations are very similar, even identical in some cases, requiring only a reversal of the key schedule [51]. Therefore the size of the code or circuitry required to implement such a cipher is nearly halved. A Feistel network is an iterated cipher with an internal function called a round function.

Let be the round function and let K0, K1, …Kn be the sub-keys for the rounds 0,1, …n respectively. Then the basic operation is as follows:

Split the plaintext block into two equal pieces, (L0, R0)

For each round I = 0, 1, ….n compute

$$L_{i+1} = R_i \quad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

$$R_{i+1} = L_i \oplus F(R_i,\ K_i) \ldots\ldots\ldots(2)$$

Then the ciphertext is (Rn+1, Ln+1)

Decryption of a ciphertext (Rn+1, Ln+1) is accomplished by computing for

I = n, n-1,…..0

$$R_i = L_{i+1} \quad \ldots\ldots\ldots\ldots\ldots(3)$$

$$L_i = R_{i+1} \oplus F(L_{i+1},\ K_i) \quad \ldots\ldots\ldots\ldots\ldots(4)$$

Then (L0, R0) is the plaintext again.

One advantage of the Feistel model compared to a substitution-permutation network is that the round function F does not have to be invertible.

Note the reversal of the subkey order for decryption; this is the only difference between encryption and decryption.

The encryption process starts with AES sub-byte transformation. The output of substitution step would be divided into two parts, namely 64-bits right part & 64-bits left part. According Feistel cipher equation 1, the right part of ith round will become left part of for i+1 round. The right part of the i+1 round is calculated by EX-Oring left part with F(Ri, Ki). The function F is discussed in following section.

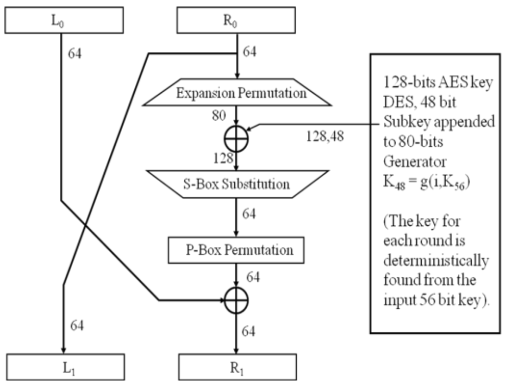
In decryption process the cipher text is processed by AES inverse substitution transformation. of substitution step would divided into two parts, namely 64-bits left part & 64-bits left part. According Feistel cipher equation 4, the right part of i+1 round will become left part of for ith round. The right part of the ith round is calculated EXOring left part with F(Ri+1, Ki+1). Here F(Ri, Ki) is a function performed on Ri+1, key Ki+1. The function F is discussed in following section.

Function F in Feistel Cipher

In F function 64-bits Right part is expanded, EX-Ored with Key, substituted using DES S-box and finally EXOred with 64-bit left part as per fesitial equation 4. Each of this operation is explained as follows:

Expansion Function

The 64-bit right half is expanded to 128-bits using expansion permutation function as shown in Figure.2. Initially 64-bits are expanded to 80-bits by dividing 64-bits into 8-subhalf of 8-bits each. In this eight bit the last & first bit is appended from previous & next 8-bit subhalf in round-robin fashion. With the generated 80-bits, DES 48-bits key is appended, to generate 128-bits.



L_0
R_0
64
64
Expansion Permutation
80
128,48
128
S-Box Substitution
64
P-Box Permutation
64
64
64
L_1
R_1
128-bits AES key DES, 48 bit Subkey appended to 80-bits Generator $K_{48} = g(i, K_{56})$
(The key for each round is deterministically found from the input 56 bit key).

Figure 2. Function F

ii) EX-Or Operation

The EX-Or operation performed on 128-bits generated from-bits last step(80-bits expanded with 64-bits DES key) is as shown in Figure3. The purpose of previous expansion was to make the right half 128-bits. The 128-bits form expansion permutation rounds are EXOred with AES 128-bits key.

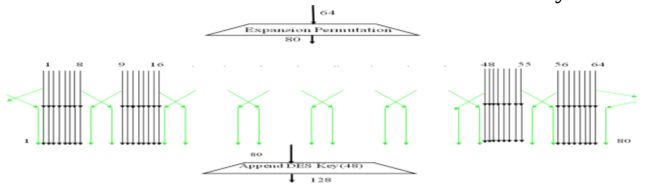


Figure 3. Substitution

Substitution

The 128-bits are converted to 64-bits using DES S-box. The operation uses sixteen S-boxes. Each box takes eight bits as input & produces 4-bits as output as shown in Figure4. The operation of individual S-box is discussed below.

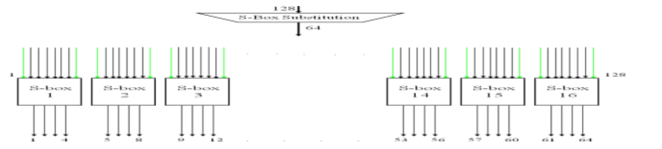


Figure 4. Permutation

Given an 8-bit input, the 4-bit output is found by selecting the row using the outer four bits (the first and last bits), and the column using the inner four bits. For example, an input "00011011" has outer bits "0011" and inner bits "0110"; the corresponding output would be "0010".

Permutation:

Figure.5 shows the P-box permutation which is nothing but the scrambling of 64-bits inputs to produce 64-bits scrambled output.

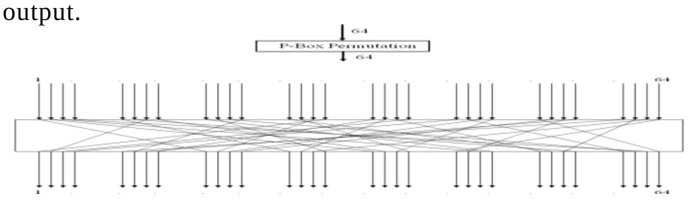


Figure 5. Permutation

Final Step: Add Round Key

The final step of the proposed algorithm is AES AddRoudKey. The AddRoundKey transformation[5] adds a round key to the state by using a simple bitwise XOR operation. Each round key is used in this transformation derived from the secret key employing the AES-key schedule is described earlier. Each round key has the same size as the state.

The encryption, decryption routines are illustrated below:

```
Procedure 1.1 Cipher(byte in[4 * Nb], byte out[4 * Nb], word w[Nb * (Nr + 1])
1: byte s[4, Nb]
```

```
2: s = in
3: SubBytes(s)
4: for round = 1 to Nr-1 do
5: Des FesitelOp(s)
6: end for
7: Add(s, w[round * Nb, (round + 1) * Nb -1])
8: out = s
```

InvCipher

Procedure 1.2 InvCipher(byte in[4 * Nb], byte out[4 * Nb], word w[Nb * (Nr + 1])

```
1: byte s[4, Nb]
2: s = in
3: InvSubBytes(s)
4: for round = Nr-1 to 1 do
5: Des Inv of FesitelOp(s)
6: end for
7: Inv of AddRound Key(s, w[round * Nb, (round + 1) * Nb -1])
8: out = s
```

Results

The proposed pseudo code for the hybrid AES-DES algorithm is implemented in Java. For the experiment, the current work has used a laptop PentiumV 2.4 GHz CPU, in which performance data is collected. The performance is analyzed for different data files. The encryption/decryption time required for specific amount of data is observed. The encryption/decryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated by dividing the total plaintext in Megabytes encrypted on the total encryption time for each algorithm in. The performance is shown in table.

Table 1. Proposed Hybrid AES Encryption & Decryption Performance

Input size in (MB)	AES(128)	Proposed Hybrid AES-DES(128)
1	0.5	0.3
50	1	0.89
100	2	1.78
128	4	3.84
200	4.5	4.1
256	5	4.7
512	9	8.66
700	11	10.56
850	12.5	12
1000	14	13.75
Average	63.5	60.58
Throughput(MB/S)	46.90	52.11

The encryption, decryption performance is graphically shown in Figure 6 . The x-axis shows the time required in seconds. The y-axis shows the size of the data files in MB. The calculated throughput and average from the observed values are also shown,

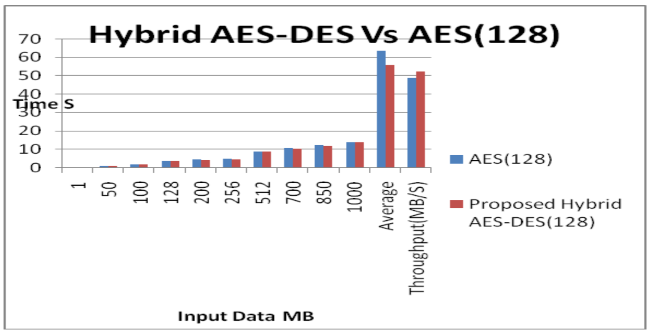


Figure 6. Performance of Proposed Hybrid AES-DES

Conclusion

In digital storage system, data security plays a crucial role and researchers from cryptographic community have spent are still spending a great deal of effort on designing and analyzing cryptographic algorithms. The proposed algorithm can be used to encrypt large data files. The algorithm is showing acceptable results as compared to standard AES (128-bit key) algorithm.

References

[1]Advanced Encryption Standard (AES), FIPS PUB 197, Nov. 26, 2001, Federal Information Processing Standards publication 1997. Federal Information Processing Standards Publication 1997.

[2] National Bureau of Standards, data Encryption Standards(DES), U S Deparatment of Commerce, Federal Information Standards Publication 46(FIPS PUB 46), 15 Jan. 1977

[3] National Institute of Standards & Technology, US Department of commerce, Commerce department announces winner of Global Information Security Competition. Available at hhp:/www.nist.govt/public_affairs/re;eases/g00-176.cf2October 2000.

[4] Vishnu, M.B.; Tiong, S.K.; Zaini, M.; Koh, S.P, Security enhancement of digital motion image transmission using hybrid AES-DES algorithm Communications, 2008. APCC 2008. 14th Asia-Pacific Conference on 14-16 Oct. 2008, Page(s): 1 – 5, Tokyo.

[5] Wang Tianfu, K. Ramesh Babu, Design of a Hybrid Cryptographic Algorithm, International Journal of Computer Science & Communication Networks,Vol 2(2), 277-283.

[6] Vikas Kaul, S K Narayankhedkar, S Achrekar, S Agarwal, P, Security Enhancement Algorithms for Data Transmission for Next Generation Networks, International Conference & Workshop on Recent Trends in Technology, (TCET) 2012, Proceedings published in International Journal of Computer Applications(IJCA).

[7] Sambhu Prasad Panda, Madhusmita Sahu, Umesh Prasad Rout and Surendra Kumar Nanda, Equivalence of DES and AES Algorithm with Cellular Automata, International Journal of Communication Network & Security, Volume-1, Issue-1, 2011.