New Approach for Modifying DES Algorithm Using 4-States Multi-keys

Dr. Rehab F. Hassan*

Received on: 14/2/2010
Accepted on:7 /10/2010

Abstract

Within the last decade, there has been a vast increase in the accumulation and communication of digital computer data in both the private and public sectors. Much of this information has a significant value, either directly or indirectly, which requires protection. One of the most important protection methods is inventing and developing different encryption algorithms. The algorithms uniquely define the mathematical steps required to transform data into a cryptographic cipher and also to transform the cipher back to the original form.

This paper introduces a new method to enhance the performance of the Data Encryption Standard (DES) algorithm. This is done by replacing the predefined XOR operation applied during the 16 round of the standard algorithm by a new operation depends on using two keys, each key consists of a combination of 4 states (0, 1, 2, 3) instead of the ordinary 2 state key (0, 1). This replacement adds a new level of protection strength and more robustness to breaking methods.

Keywords: computer security, data encryption standard, triple data encryption algorithm.

طريقة جديدة لتحسين خوارزمية التشفير DES باستخدام مفاتيح متعددة متكونة من اربع حالات

الخلاصة

في العقد الاخير صار هناك تزايد في تراكم وتواصل بيانات الحاسوب الرقمي في كلا الاتجاهين العام والخاص. معظم هذه البيانات تكون ذات قيمة بشكل مباشر او غير مباشر وبهذا فهي تتطلب الحماية. واحدة من اهم طرق الحماية هو اكتشاف وتطوير مختلف خوارزميات التشفير. هذه الخوارزميات تعرف بشكل موحد الخطوات الرياضية المطلوبة لتحويل البيانات الى شفرة سرية وايضا لتحويل هذه الشفرة الى شكلها الاصلي.

هذا البحث يقدم طريقة جديدة لتعزيز اداء خوارزمية DES. وذلك بابدال عملية XOR المطبقة في 16 دورة في الخوارزمية القياسية بعملية جديدة تعتمد على استخدام مفتاحين كل منهما يتكون من اربع حالات وهي (0, 1, 2, 3) بدلا من استخدام المفتاح الاعتيادي والذي يتكون من حالتين وهما (0, 1).هذا التبديل يضيف مستوى جديد من قوة الحماية و قوة الممناعة لطرق الكسر.

1. Introduction

Cryptography has a long and fascinating history. Beginning with the work of Feistel at IBM in the early 1970s and culminating in 1977 with the adoption as a U.S. Federal Information Processing Standard for encrypting unclassified information, DES, the Data Encryption Standard, is the most well-known

* Computer Science Department, University of Technology / Baghdad

https://doi.org/10.30684/etj.28.20.8
2412-0758/University of Technology-Iraq, Baghdad, Iraq

cryptographic mechanism in history. It remains the standard means for securing electronic commerce for many financial institutions around the world [4, 5].

The most striking development in the history of cryptography came in 1976 when Diffie and Hellman published New Directions in Cryptography. This paper introduced the revolutionary concept of public-key cryptography and also provided a new and ingenious method for key exchange, the security of which is based on the intractability of the discrete logarithm problem. Although the authors had no practical realization of a public-key encryption scheme at the time, the idea was clear and it generated extensive interest and activity in the cryptographic community [1, 2].

In 1978 Rivest, Shamir, and Adleman discovered the first practical public-key encryption and signature scheme, now referred to as RSA. The RSA scheme is based on another hard mathematical problem, the intractability of factoring large integers. This application of a hard mathematical problem to cryptography revitalized efforts to find more efficient methods to factor. The 1980s saw major advances in this area but none which rendered the RSA system insecure. Another class of powerful and practical public-key schemes was found by ElGamal in 1985. These are also based on the discrete logarithm problem [1,3].

The search for new public-key schemes, improvements to existing cryptographic mechanisms, and proofs of security continues at a rapid pace. Various standards and infrastructures involving cryptography are being put in place. Security products are being developed to address the security needs of an information intensive society [7]. The research is an attempt to improve most cryptographic algorithms that depend on using logical OR operation, then the DES algorithm is explained briefly in section 3, where the proposed improvement will be applied on, section 4 contains a full description to the operation that will be exchanged with the ordinary OR operation. The rest of the paper explains the new improved DES algorithm, with the conclusion and the suggested future work that can be applied.

2. Data Encryption Standard (Des)

Without doubt the first and the most significant modern symmetric encryption algorithm is that contained in the Data Encryption Standard (DES) [2]. 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). The algorithm has been in wide international use,

The Data Encryption Standard (DES), as specified in FIPS Publication 46- 3 [5], is a block cipher operating on 64-bit data blocks. 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 (1). The decryption process is the same as the encryption, except for the order of the round keys used in the Feistel iterations [4].

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 L_0 and R_0). In each iteration (or round), the second word Ri is fed to a function f and the result is added to the first word Li . Then both words are swapped and the algorithm proceeds to the next iteration. The function f is key-dependent and consists of four stages [4]:

1. Expansion (*E*). The 32-bit input word is first expanded to 48 bits by duplicating and reordering half of the bits.

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 RBlock is then XORED with LBlock and the resultant fed to the next RBlock register. The unmodified RBlock is fed to the next LBlock register. With another 56 bit derivative of the 64 bit key, the same process is repeated. Full details of DES are given in Algorithm (1) [3].

3. Improved 4-States operation

To increase the security and key space, that makes the encryption algorithms more robustness to the intruders, a new manipulation bits process has been added in [6] by using different truth table for manipulation bits process work on 4-states (0,1,2,3) , while the traditional binary process (XOR) work on (0, 1) bits only. The symbol # has been used to refer to the operator that execute this process use truth tables that shown in figure (2) [6].

The new operation needs 3 inputs, the first one specify the table number that should be used to calculate the result among the 4 tables, the other 2 inputs define the row and column number in the specified table where the cross point of them gives the result. An example of applying the operation is shown below:

Input 1: 0 1 3 0 1 2 2 3 1
Input 2: 3 2 2 1 0 1 2 1 1
Input 3: 1 0 0 2 1 3 2 1 2

Result : 1 2 0 0 1 2 2 3 2

4. The proposed algorithm to modify DES using 4-states

This research proposed a new improvement to the DES algorithm. The proposed improvement makes use of the new operation defined in the previous section, operation (#) applied during each round in the original DES algorithm, where another key is needed to apply this operation, this key may come in binary form and convert to a 4-states key, or it may already come in a 4-states as that can be done with quantum channel.

Consequently, multiple keys will be used in each round of the original DES, the first key Ki will be used with the f function. The second key will be the first input to the # operation, the second input will be the output of the f function, and the third input to the # operation will be the value Li, Figure (3) shows the three 32-bits input to the # operation , and the 32-bits output, with the

places needed to convert these 32-bits to 16-digits. These three inputs to the # operation should be firstly converted from 32 bits to a 16 digits each may be one of four states (0, 1, 2, 3), i.e., each two bits converted to its equivalent decimal digits, for example, the binary number:

10010111010100101010011110100010001

will be converted to the number:

2 1 1 3 1 1 0 2 2 2 1 3 2 2 0 2 1

Then the # operation will be applied to generate a new 16 digits that should be reconverted to 32 bits to be the input to Ri+1, see Figure (4). Full details of the proposed improved DES are given in Algorithm (2) [6].

5. Implementation

The following example shows how the encryption and decryption operations results will be according to apply the # operation, so in any stage we can expect that the result of the previous left part of the data could be the binary number:

Li-1= 00101010010101011101010010111101110 0

And the result of the function ***f***, which represents here the first key, could be the binary number:

***f*=10010100011101010101001001110100110 1**

the entered key, which represent the second key in the applied # operation, the binary number:

ki'=11101010010101011010111010101001011 11

Firstly, the three entered 32-bits binary numbers should be converted to a 4-states 16-digits numbers:

***f* '= 2 1 1 0 1 3 1 1 1 1 0 2 1 3 1 0 3 1**
Li-1'= 0 2 2 2 1 1 1 1 3 1 1 0 2 3 3 1 3 0
ki''= 3 2 2 2 1 1 1 1 2 2 3 2 2 2 2 1 1 3

Then the # operation applied according to tables shown in figure (2) , the result of encryption will be:

Ri'= 3 1 1 0 0 2 0 0 0 2 0 0 1 2 0 1 0 0

If we reverse the whole operation we will get the initial number, which is the result of the encryption operation that equal to the original data:

Ki''= 3 2 2 2 1 1 1 1 2 2 3 2 2 2 2 1 1 3
***f* '= 2 1 1 0 1 3 1 1 1 1 0 2 1 3 1 0 3 1**
Ri'= 3 1 1 0 0 2 0 0 0 2 0 0 1 2 0 1 0 0

Li-1'= 0 2 2 2 1 1 1 1 3 1 1 0 2 3 3 1 3 0
Li-1= 00101010010101011101010010111101110 0

6. Conclusions

DES is now considered to be insecure for many applications. This is chiefly due to the 56-bit key size being too small; in January, 1999, distributed.net and the Electronic Frontier Foundation collaborated to publicly break a DES key in 22 hours and 15 minutes There are also some analytical results which demonstrate theoretical weaknesses in the cipher, although they are unfeasible to mount in practice. So it becomes very important to augment this algorithm by adding new levels of security to make it applicable and can be depending on in any common communication channel. Adding additional key and replacing the old XOR by a new operation as proposed by this paper to give more robustness to DES algorithm and make it stronger against any kind of intruding. The ciphering process stills simple and can be implemented by hardware in this new proposed improvement, as well as the time complexity of the new algorithm stays the same since only one operation is replaced by another operation, and the conversion operations is very simple and straightforward.

7. References

[1] Alan G. Konheim, "COMPUTER SECURITY AND CRYPTOGRAPHY ",2007 , by John Wiley & Sons, Inc.

[2] Alfred J.M., Paul V. C. and Scott A. V., "Handbook of Applied Cryptography", Fifth Addition, 2001.

[3] Bruce Schneier, "Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C", 1996, Wiley Computer Publishing, John Wiley & Sons, Inc.

[4] Coppersmith, Don. (1994). "The data encryption standard (DES) and its strength against attacks". IBM Journal of Research and Development, 38(3), 243–250.

[5] National Institute of Standards and Technology, (1979). "FIPS-46: Data Encryption Standard (DES)." Revised as FIPS 46-1:1988, FIPS 46-2:1993, FIPS 46-3:1999, available at http://csrc.nist7az nml;' .gov/publications/fips/fips46-3/fips46-3.pdf

[6] Hala Bahjat AbdulWahab1 , Abdul Monem S. Rahma, 'Proposed New Quantum Cryptography System Using Quantum Description techniques for Generated Curves", The 2009 International conference on security and management, SAM2009, July 13-16 2009, Las Vegas, USA, SAM 2009.

[7] Henk C.A. van Tilborg, Eindhoven , "ENCYCLOPEDIA OF CRYPTOGRAPHY AND SECURITY", 2005, Springer Science+Business Media, Inc.

Algorithm (1): Data Encryption Standard (DES)

INPUT : plaintext m1 . . . m64; 64-bit key K=k1 . . . k64 (includes 8 parity bits).
OUTPUT : 64-bit ciphertext block C=c1 . . .c64.
1. (key schedule) Compute sixteen 48-bit round keys Ki, from K.
2. (L0, R0) ¬ IP(m1, m2,. . .m64) (Use IP Table to permute bits; split the result into left and right 32-bit halves L0=m58m50 . . . m8,R0=m57m49 . . . m7)
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 Å *f* (R i-1, Ki) where *f*(Ri-1, Ki) = P(S(E(Ri-1) Å Ki)), computed as follows:
(a) Expand Ri-1 = r1r2 . . . r32 from 32 to 48 bits, T ¬ E(Ri-1).
(b) T' ¬ T Å Ki . Represent T ' as eight 6-bit character strings: T '= (B1 . . . B8)
(c)T '' ¬ (S1(B1), S2(B2), . . . S8(B8)). Here Si(Bi) maps to the 4-bit entry in row r and column c of Si
(d)T''' ¬ P(T''). (Use P per table to permute the 32 bits of T''=t1t2 . . . t32, yielding t16t7 . . . t25.)
4. b1b2 . . . b64 ¬ (R16, L16). (Exchange final blocks L16, R16.)
5. C ¬ IP-1 (b1b2 . . . b64).
6. End.

Algorithm (2): The modified Data Encryption Standard (DES) using 4-states operation

INPUT : plaintext m1 . . . m64; 64-bit two keys K=k1 . . . k64 and K'=k1' . . . k64' (includes 8 parity bits).
OUTPUT : 64-bit ciphertext block C=c1 . . .c64.
1. (key schedule) Compute sixteen 48-bit round keys Ki, from K.
2. (key schedule) compute sixteen 32-bit round keys Ki', from K'
2. (L0, R0) ¬ IP(m1, m2,. . .m64) (Use IP Table to permute bits; split the result into left and right 32-bit halves L0=m58m50 . . . m8,R0=m57m49 . . . m7)
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 # *f* (R i-1, Ki) where *f*(Ri-1, Ki) = P(S(E(Ri-1) Å Ki)), computed as follows:
(a) Expand Ri-1 = r1r2 . . . r32 from 32 to 48 bits T ¬ E(Ri-1). (Thus T= r32r1r2 . . . r32r1.)
(b) T' ¬ T Å Ki . Represent T ' as eight 6-bit character strings: T ' = (B1 . . . B8)
(c)T '' ¬ (S1(B1), S2(b2), . . . S8(B8)). Here Si(Bi) maps to the 4-bit entry in row r and column c of Si
(d)T''' ¬ P(T''). (Use P per table to permute the 32 bits of T''=t1t2 . . . t32, yielding t16t7 . . . t25.)
and the operation # in Ri = Li-1 #*f* (R i-1, Ki) is computed as follows:
(a) convert the 32 bits resulted from*f* (R i-1, Ki) into 4-states 16 digits call it*f* '
(b) convert the 32 bits of Li-1 to 4-states 16 digits call it Li-1'
(c) convert the 32 bits of Ki' to 4-states 16 digits call it Ki''
(d) compute Ri by applying the # operation on*f* ', Li-1', and Ki'' according to truth tables shown in figure (2)
4. b1b2 . . . b64 ¬ (R16, L16). (Exchange final blocks L16, R16.)
5. C ¬ IP-1 (b1b2 . . . b64). (Transpose using IP-1 C = b40b8 . . . b25.)
6. End.

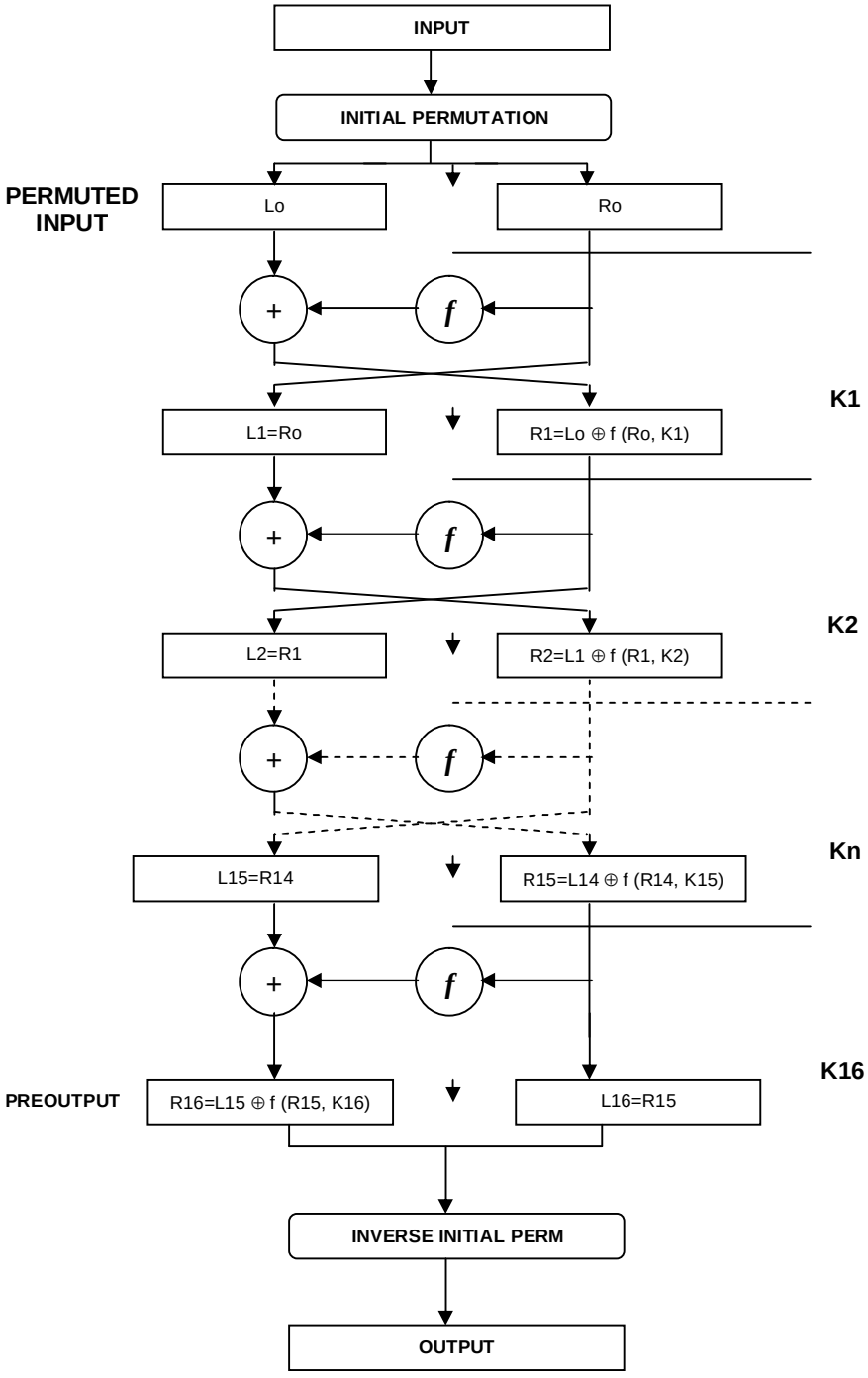

Figure (1): DES computation path.

#0	0	1	2	3
0	3	2	1	0
1	2	3	0	1
2	1	0	3	2
3	0	1	2	3

#1	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

#2	0	1	2	3
0	2	3	0	1
1	3	2	1	0
2	0	1	2	3
3	1	0	3	2

#3	0	1	2	3
0	10	1	0	3	2
1	01	0	1	2	3
2	32	3	2	1	0
3	23	2	3	0	1

Figure (2) The truth tables for the # operation

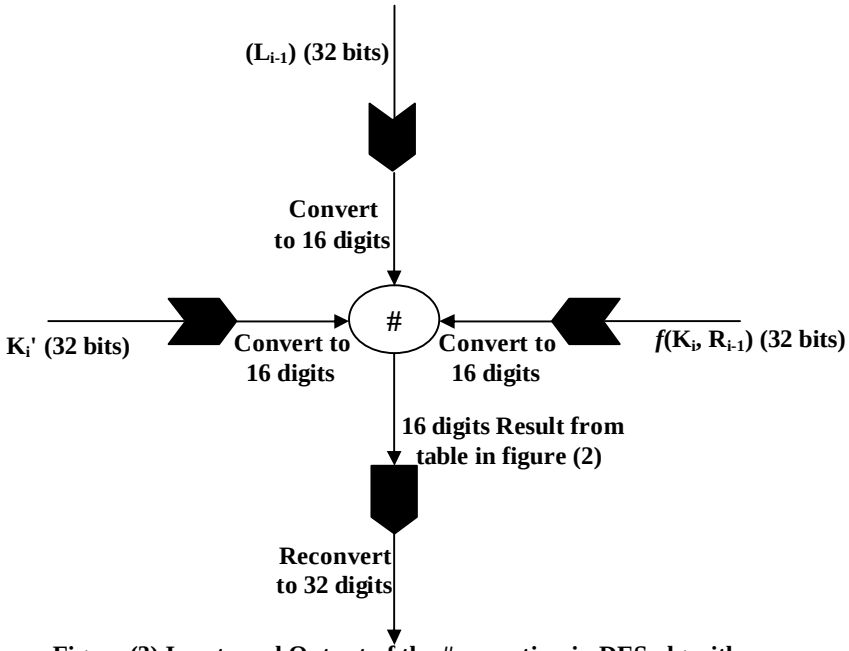

Figure (3) Inputs and Output of the # operation in DES algorithm

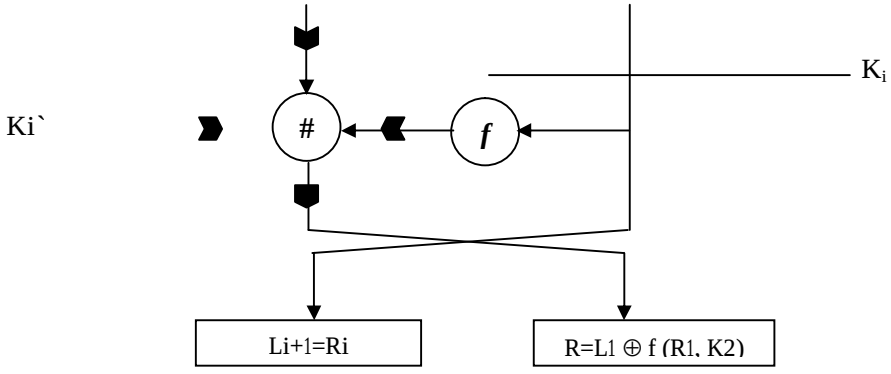

Figure (4) The new structure of each round in the improved DES Algorithm