Modified Key Model of Data Encryption Standard

Salih Mohammed Salih
IEEE Member, Electrical Department
College of Engineering, University of Anbar, Iraq
E-Mail: dr_salih_moh@yahoo.com

Abstract:
This paper specifies a proposed improvement model of Data Encryption Standard (DES) which may be used to protect sensitive data. Protection of data during transmission may be necessary to maintain the confidentiality and integrity of the transformation represented by data. Instead of expansion step in each round which made by copying 16 bit from 32 bits data in each right side of the standard algorithm, the unused 8-bits as a key (*sometimes it is used for error detection and correction purposes*, *or it is possible to generate an additional 8-bits with the 56-bits standard key*) in the first starting round with the other 8-neglected bits from each of 16 round in the key algorithm will be used, and take the same locations of the expanded data. As a result, the complexity to cryptanalysis of the secured data has been increased. The proposed method was more active and reliable than standard conventional DES, where it can be switched to the system at any round for working with original DES algorithm, which means that an additional security has been added.

Keywords: ***DES, S-Box, Key Generation, Permutation, Rounds***

1. INTRODUCTION

The selective application of technological and related procedural safeguards is an important responsibility of every organization in providing adequate security to its electronic data systems. Protection of data during transmission or while in storage may be necessary to maintain the confidentiality and integrity of the information represented by the data. The algorithm uniquely defines the mathematical steps required to transform data into a cryptographic cipher and also to transform the cipher back to the original form. Data encryptions standard (DES) use 64 bits block size as well as 64 bits key size that are vulnerable to bruteforce, attack. But for both efficiency and security, a larger block size is desirable [1]. The Advanced Encryption Standard (AES,) that uses 128 bit block size as well as 128 bits key size was introduced by NIST [2].

The most widely used encryption scheme is based on the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46). The algorithm itself is referred to as the Data Encryption Algorithm (DEA). For DES, data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption [3]. The terminology is a bit confusing, because recently, the terms DES and DEA could be used interchangeably. However, the most recent edition of the DES document includes a specification of the DEA plus the triple DEA (TDEA). Both DEA and TDEA are part of the Data Encryption Standard. Further, until the recent adoption of the official term TDEA, the triple DEA algorithm was typically referred to as triple DES and written as 3DES, for the sake of convenience [4, 5].

In the late 1960s, IBM set up a research project in computer cryptography led by Horst Feistel [6]. The project concluded in 1971 with the development of an algorithm with the designation Lucifer [7], which was sold to Lloyd's of London for use in a cash-dispensing

system, also developed by IBM. Lucifer is a Feistel block cipher that operates on blocks of 64 bits, using a key size of 128 bits. Because of the promising results produced by the Lucifer project, IBM embarked on an effort to develop a marketable commercial encryption product that ideally could be implemented on a single chip.

In 1973, the National Bureau of Standards (NBS) issued a request for proposals for a national cipher standard [8]. IBM submitted the results of its Tuchman-Meyer project. This was by far the best algorithm proposed and was adopted in 1977 as the Data Encryption Standard. Before its adoption as a standard, the proposed DES was subjected to intense criticism, which has not subsided to this day. Two areas drew the critics' fire. First, the key length in IBM's original Lucifer algorithm was 128 bits, but that of the proposed system was only 56 bits, an enormous reduction in key size of 72 bits. Critics feared that this key length was too short to withstand brute-force attacks. The second area of concern was that the design criteria for the internal structure of DES, the S-boxes, were classified. Thus, users could not be sure that the internal structure of DES was free of any hidden weak points that would enable NSA to decipher messages without benefit of the key. Subsequent events, particularly the recent work on differential cryptanalysis, seem to indicate that DES has a very strong internal structure. Furthermore, according to IBM participants, the only changes that were made to the proposal were changes to the S-boxes, suggested by NSA, that removed vulnerabilities identified in the course of the evaluation process. In 1994, NIST recommended the use of DES for applications other than the protection of classified information. In 1999, NIST issued a new version of its standard (FIPS PUB 46-3) that indicated that DES should only be used for legacy systems and that triple DES (which in essence involves repeating the DES algorithm three times on the plaintext using two or three different keys to produce the ciphertext) is used [3].

Mehran M. and Arash Reyhani M. [2] have studied a number of fault detection schemes for the encryption and the decryption of the Advanced Encryption Standard (AES). New fault detection schemes which are independent of the structures of the S-boxes and the inverse S-boxes have been proposed. Their simulations show that for the AES encryption and decryption, these structure-independent schemes reach the error coverage of approximately 100%. Many researchers have researched for new modified models of encryption and decryption to make the system more complex for attackers. Other papers are concentrated on cryptanalysis to break the key, whether to get the data either in a passive or active attack. In this paper, we suggested another modified algorithm by using the DES Algorithm to enhance its performance and make it more difficult to the attackers for breaking the key and extract the useful data.

2. PROPOSED DES ALGORITHM

The algorithm is designed to encipher and decipher a block of data consisting of 64-bits under control of a 64-bit key [8]. Deciphering must be accomplished by using the same key as the enciphering side, but with a schedule of addressing the key bits altered so that deciphering process is the reverse of the enciphering process. In this algorithm we don't make an expansion of right side of data, but on the other hand we use the deleted 8-bits in each round in the key. The deleted 8-bits in each round are put in place of the expansion of data such that a 32-bit of data will be 40-bit in addition to 8-bits of key that will be deleted in the first round of key, this will make the total key used with data in each round of 16-bit, so the total value in the expansion step will be 48-bit as is standard. The proposed improvement model depends on using the key in each round instead of copying the data from the right side in each round of 16 bit to expand it, such that the total number of bits will be 48 bits.

The first step for ciphering the data is the initial permutation (IP) F**ig. (1)**, secondly to a complex key-dependent computation and finally to a permutation which is the inverse of the

initial permutation (IP^{-1}). The 64-bit of the input block to be enciphered is first subjected to the following permutation called the initial permutation (IP) according to the following standard locations for each bit. The output of this peroutput computation is then subjected to the inverse initial permutation shown in F**ig. (2)**.

For our work we will depend on the standard algorithm by using DES 64-bit with the same S-Box and Key generation. The changing in algorithm will be done by exchanging the function of expansion permutation (E) by using other function that will be addressed as distribution of (D-K), it consists of 16-bit that deleted from permutation choice one (PC-1) and permutation choice two (PC-2) in the key, and as shown in **Fig. (3)**

3. DETAILS OF A SINGLE ROUND

By focusing on the left hand side of the diagram in **Fig. (4)**, the left and right halves of each 64-bit intermediate value are treated as separate 32-bit, its 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:

$Li=Ri-1$, $Ri=Li-1 \oplus F(Ri-1,Ki)$

Fig. (5) shows the block diagram of a single round of DES Algorithm. In this figure, the data on the right side (32-bit) have been expanded to 48-bit, then it is correlated by XOR-gate with a 48-bit of key. While in **fig. (5)** we used the key instead of the expanded bits in each round.

4. A DEMONSTRATION OF NUMERICAL EXAMPLE OF THE FIRST ROUND

Let us consider an input random data of 64-bit is to be used by the suggested model of DES as given below:

$$Data = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & & & & & & & & & & & & & & & & \end{bmatrix}$$

Let us take input key as:

$$Key = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & & & & & & & & & & & & & & & & \end{bmatrix}$$

The next step is the initial permutation

$$\text{Initial permutation} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Round (1):
Left side of data after initial permutation

$$L_0 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Right side of data after initial permutation

$$R_0 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\text{Framing (K - D)1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Key permutation choice two} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

F=framing (K-D)**XOR** (PC-1)

$$F = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{S - BOX} = \begin{bmatrix} 0\,1\,0\,1 \\ 0\,1\,1\,0 \\ 0\,0\,0\,1 \\ 1\,1\,1\,1 \\ 0\,1\,1\,0 \\ 1\,1\,0\,1 \\ 0\,0\,0\,0 \\ 1\,0\,1\,1 \end{bmatrix}$$

The bits distributions of the Permutation function as in **Fig. (6)**, the output from S-Box will permute as in the next equation:

$$\text{Permutation} = \begin{bmatrix} 1\,1\,0\,1\,1\,1\,0\,0 \\ 0\,1\,0\,0\,0\,1\,1\,0 \\ 1\,0\,1\,1\,1\,0\,0\,0 \\ 1\,1\,0\,1\,1\,0\,1\,0 \end{bmatrix}$$

Right 1= (L_0) **XOR** (P)

$$R1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$L_1=R_0$

$$L1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

This is round one from 16 rounds, the remainder rounds will be implementing as the first round, and the final results for end round will be as:
Round (16):

$$R16 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$L16 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The out algorithm before inverse initial permutation will be:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

The inverse permutation will be:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The data transmitted will be as following:

$$\text{Enciphered data} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

5. PERFORMANCE EVALUATION OF THE PROPOSED DES

The DES with a random 64-bit of plaintext was tested with random 64-bit of key. The results of each round of the enciphered data was compared with the input plaintext to see the bit difference in each round relative to the assumed input binary data. These results are given in **Table (1).** In the same table we wrote the results of the conventional of DES. From this table, it can be seen that more bit differences have been obtained from the suggested model. Both of the algorithms started at 28-bit difference in the first round. The second round gives an increasing 8-bit difference relative to the previous round even the standard DES be

constant at 28-bit. In all rounds there are 558-bit difference be obtained from the proposed model compared with 540-bit difference of the conventional DES this means, the security of data has been increased by about 4%. Another issue is related with the ability of using the new method with the standard DES algorithm, is because we kept the blocks of DES without variations. The switching between the two Algorithms can be taken as an encryption key in addition to the standard used key.

Another advantage can be obtained from the suggested model related to increasing the time required to Attack the plain text. Where in the DES it can be Attack the data if the 56-bit key is known. While in our algorithm the Attacker must spend additional time to decrypt the other 8-bits that have been used in the proposed model. This means the security of the algorithm was increased by (8/64) i.e.12.5% in addition to the 4% that was obtained from the algorithm.

The other issue related with the ability of using the DES algorithm and the suggested one in the same time. Or in other form it can be switched between the two systems according to special key or with using the first 16 bit of the generated key. For example, if the first 16 bits are: [1 0 0 0 1 0 0 0 1 1 1 0 1 0 1 0], then we can consider each logic'1' of this key represents the system that will be switched to the new algorithm in the same round that represented by the bit location. For this 16 bit of key; the system will be switched into the proposed model in rounds [1, 5, 9, 10, 11, 13, and 15] which represents the locations of ones. This case will cause an increasing of data security by a time required to cryptanalysis of 16! (Factorial of 16-rounds).

Hint: in the previous results we assumed the first 8-bit from the total 64-bit key as an input the first round and it is used as a shared key for all rounds but sometimes these bits were used for error detection and correction so it is not a condition to use these bits in the suggested algorithm.

6. THE 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 ciphertext. In particular, a change in one bit of the plaintext should produce a change in many bits of the ciphertext. If the change were small, this might provide a way to reduce the size of the plaintext space to be searched. DES exhibits a strong avalanche effect. **Table (2)** shows some results taken from using two plaintexts that differ by one bit were used:

$$\text{Assumed plaintext before changing} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Plaintext differs in only one bit position (first bit) =

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

7. CONCLUSIONS

The alternative improvement model was introduced and tested. The results show that the suggested model has the ability to increase the security by about 4% due to the using of key in the same locations of the expanded data in each round. A 12.5% increasing of security was also obtained by using the 64-bit key instead of 56-bit used in the standard model. The blocks of DES model and the proposed model are the same and without addition of new blocks, also it can be switched between the proposed and the standard model at any round according to a secure key used for switching between models. The last case increased the time required to attack the algorithm because we have 16! case for switching between the two models.

8. REFERENCES

[1] Md. Nazrul Islam, Md. Monir Hossain Mia, Muhammad F. I. Chowdhury, and M.A. Matin, "Effect of Security Increment to Symmetric Data Encryption through AES Methodology," Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 291-294, (2008)

[2] Mehran Mozaffari-Kermani and Arash Reyhani-Masoleh, "Concurrent Structure-Independent Fault Detection Schemes for the Advanced Encryption Standard" accepted for publication in IEEE Transactions on Computers-(2010), Copy right is 0018-9340/10/$26.00 © 2010 IEEE.

[3] William Daley. Raymond G. kammer, "Data Encryption Standard," U.S. Department of Commerce/ National Institute of Standards and Technology, October, (1999).

[4] Vano90 van Orschot, P., and Wiener, M., "A Known-Plaintext Attack on Two-Key Triple Encryption," Proceedings, EUROCRYPT 90, published by Springer-Verlag, (1990).

[5] Jing Wang, and Guo-ping Jiang, "Improved DES Algorithm Based on Irrational Numbers," IEEE International Conference Neural Networks and Signal Processing. China, June 8-10, pp. 632-635 (2008)

[6] Bish05 Bishop, M., "Introduction to Computer Security," Boston: Addison-Wesley, (2005).

[7] Feis75 Feistel, H. Notz, W. and Smith, J., "Some Cryptographic Techniques for Machine-to-Machine Data Communications," Proceedings of the IEEE, November (1975).

[8] Pfleeger, C., "Security in Computing, Upper Saddle River," NJ: Prentice Hall, (2002).

[9] Li Juan, Chen Bin, and Li Kun, "Study on the Improvement of Encryption Algorithm of Bluetooth," International Conference on Networking and Digital Society. pp. 89-92, (2009).

[10] Coppersmith, D., "The Data Encryption Standard (DES) and Its Strength Against Attacks," IBM Journal of Research and Development, May (1994).

[11] Electronic Frontier Foundation. "Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design," Sebastopol, CA: O'Reilly, (1998).
[12] Menezes, A., van Oorschot, P., and Vanstone, S., "Handbook of Applied Cryptography," Boca Raton, FL: CRC Press, (1997).
[13] Schneier, B., "Applied Cryptography," New York: Wiley, 1996.
[14] Simovits, M., "The DES: An Extensive Documentation and Evaluation," Laguna Hills, CA: Aegean Park Press, (1995).
[15] Stinson, D., "Cryptography: Theory and Practice," Boca Raton, FL: CRC Press, (2002).

Table (1): The bit differences in each round relative to the input data

Number of round	*Number of changes in the output data for standard DES*	*Number of changes in the out data for propose mode*
0	28	28
1	28	36
2	32	34
3	36	36
4	32	32
5	28	28
6	32	33
7	36	36
8	32	32
9	28	28
10	32	32
11	36	36
12	32	32
13	28	28
14	32	39
15	36	36
16	32	32

Table (2): Avalanche effect of the proposed DES

Number of round	*Plaintext with Change Number of bits that differ*	*Number of round*	*Plaintext without Change*
0	*27*	*0*	*28*
1	*32*	*1*	*36*
2	*32*	*2*	*34*
3	*35*	*3*	*36*
4	*39*	*4*	*32*
5	*30*	*5*	*28*
6	*32*	*6*	*33*
7	*34*	*7*	*36*
8	*33*	*8*	*32*
9	*34*	*9*	*28*
10	*39*	*10*	*32*
11	*35*	*11*	*36*
12	*32*	*12*	*32*
13	*30*	*13*	*28*
14	*32*	*14*	*39*
15	*36*	*15*	*36*
16	*32*	*16*	*32*

(a) Initial Permutation (IP)							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Figure (1): Bits locations of the initial permutation function

(b) Inverse Initial Permutation (IP^{-1})							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Figure (2): Bits locations of the inverse initial permutation function

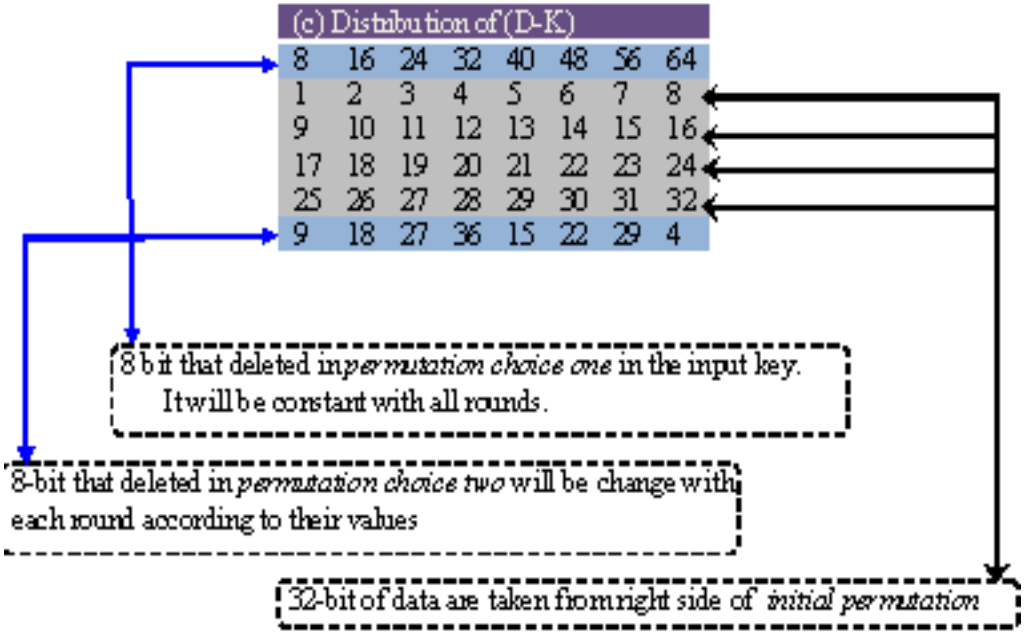

Figure (3): Distribution of data and keys in the modified model.

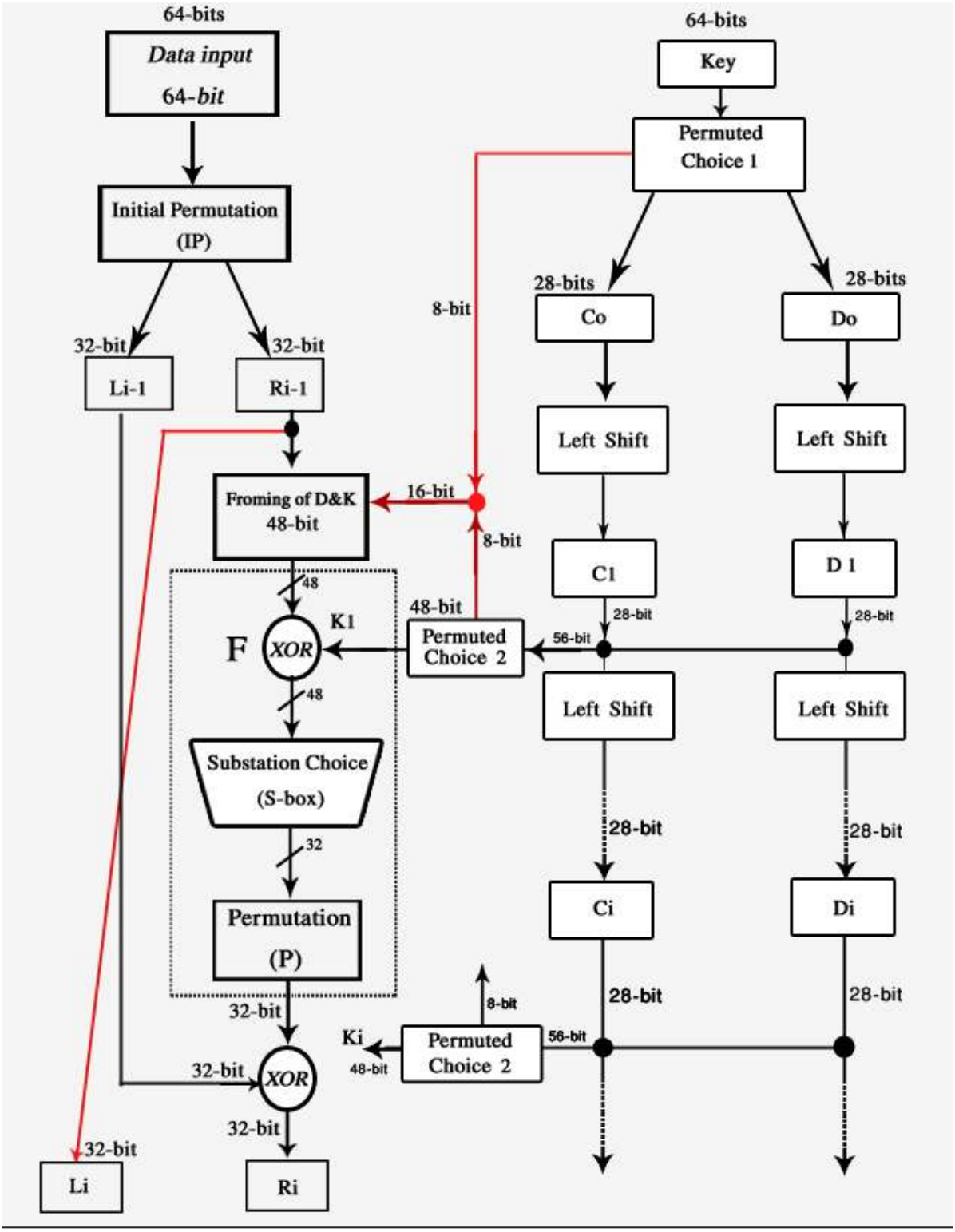

Figure (4): Single Round of the Proposed DES Algorithm

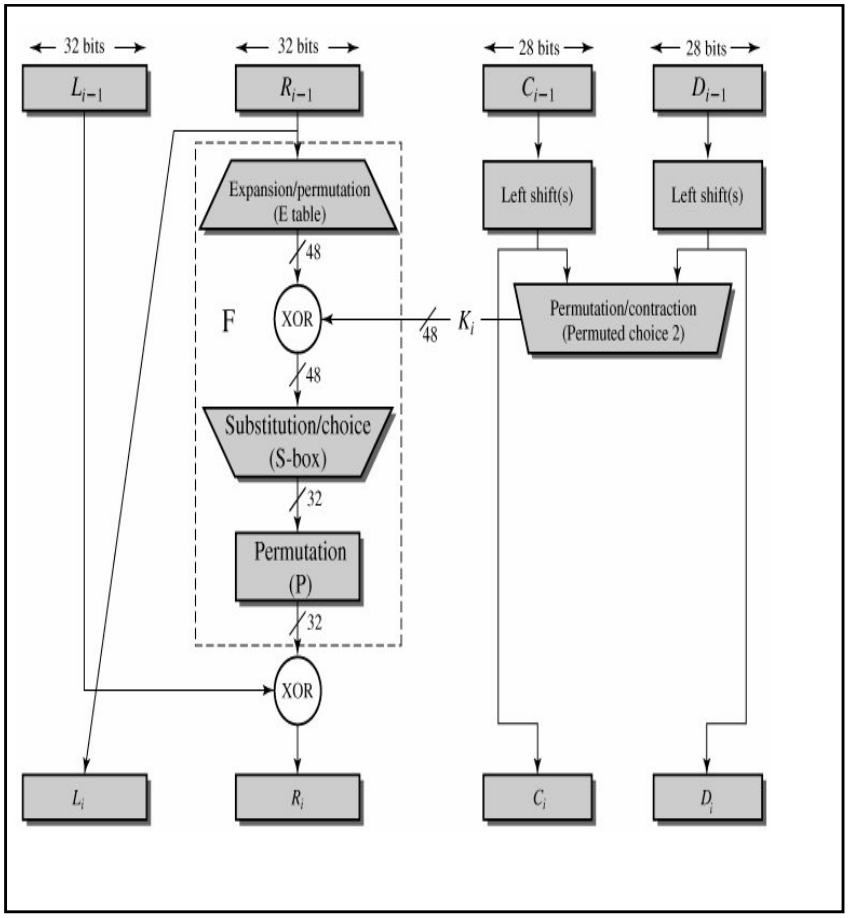

Figure (5): Single Round of DES Algorithm

(d) Permutation Function (P)							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Figure (6): Bits distribution of the permutation function

نموذج بديل ومحسن لتشفير البيانات القياسي

د. صالح محمد صالح
جامعة الانبار/ كلية الهندسة/ قسم الهندسة الكهربائية

الخلاصة:

في هذا البحث تم اقتراح نموذج اخر لتشفير البيانات معتمدا على الخوارزمية الاصلية والقياسية المستخدمة لحماية البيانات الحساسة والهامة. ان حماية البيانات خلال عملية النقل تكون ضرورية للمحافظة على وثوقية وسلامة ارسالها. حيث تم الغاء خطوة استنساخ البيانات وتوسيعها ببداية كل دورة للجانب الايمن من البيانات (32 بت) والتي تتوسع مع الاستنساخ لتكون 48 بت باستبدالها بالبتات الثمانية الاولى من المفتاح المشفر (عادة تستخدم البتات من 57–64 لاغراض كشف وتصحيح الخطا، أو بالامكان توليد 8 بتات اضافية للدورة الاولى) مع البتات الثمانية الاخرى المهملة من كل دورة من الدورات الستة عشر ليكون لدينا 16 بت اضافي من كل دورة، توضع هذه البتات بدل من موقع البتات المستنسخة بالنظام القياسي عند كل دورة من الدورات الستة عشر. النتائج بينت ان كسر الشفرة اصبح اكثر تعقيدا وهذا يعني ان الطريقة المقترحة ذات فعالية اكبر بالاضافة الى مرونتها، حيث بالامكان استخدام النظام القياسي او المقترح في اي دورة من الدورات الستة عشر، وبالتالي اضافت امن للبيانات بشكل اكبر.