Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

7th International Conference on Electrical Engineering
ICEENG 2010

A customized 152 bit key DES algorithm

By

Brig.Gen.Prof.Dr/Salah S. El Agooz*

Col.Ass.Prof.Dr/Alan Eldin R. Shehata*
alaa_rohiem@yahoo.co.uk

Col.Dr/Essam Abd Elwanees Amer *
mohwamees@yahoo.com

Capt.Eng/ Mahmoud Y.Ahmed**
m.yeh_82@hotmail.com

Abstract:

In this paper, we proposed a novel technique for modification of the Data Encryption Standard (DES) to ensure a security enhancement against cryptographic attacks that were the main problem in DES algorithm. The proposed modification extended the key size from 64 bit to 152 bit, these extra bits are added to make the S-boxes be dynamically instead of the standard DES where it has static S-boxes, In this paper two approaches are introduced for Algorithm modification. The paper also includes a performance evaluation comparison between the proposed algorithm and the standard DES at different settings for data sizes, data type and encryption/decryption speed. The results show that the proposed techniques for algorithm modification make the DES algorithm stronger against cryptographic attacks compared to standard DES algorithm.

Keywords:

Security, DES evaluation, encryption time, throughput.

---

* Military Technical College, Cairo, Egypt
** Egyptian Armed Forces

1. Introduction:

The **Data Encryption Standard** (**DES**) is a ciphering algorithm selected as an official Federal Information Processing Standard. Federal Information Processing Standards FIPS are publicly announced standards developed by the United States Federal government for use by all (non-military) government agencies and by government contractors. The algorithm was initially controversial, with classified design elements, a relatively short key length. DES consequently came under intense academic scrutiny, and motivated the modern understanding of block ciphers and their cryptanalysis [1-3].
Due to the small key length of DES and simple Feistel network, many cryptanalysts developed various methods to break, where it is most vulnerable to differential and linear cryptanalysis attack [5-6]. DES is considered insecure against the brute force attack where a highly speed development processors are employed now.
The DES algorithm has a certain theoretical weaknesses; therefore it is infeasible to mount in practice. In recent years, the cipher has been superseded by the Advanced Encryption Standard (AES) [1]. To explain the proposed algorithm, at first we show how DES works and the commonly used cryptanalysis techniques to break the DES. The paper is organized as follows. The DES algorithm is described in Section 2, cryptanalysis of DES are shown in section 3, The proposed DES algorithm is illustrated in section 4,the performance evaluation of the proposed DES and the standard DES and Simulation results are shown in section 5 and The conclusions are shown in section 6.

2. The DES algorithm:

The overall scheme for the encryption process is illustrated in Figure 1, where the plaintext is divided into blocks with 64 bits to be encrypted in three phases. Firstly, it passes through an initial permutation (IP) and followed by a phase consists of identical (16) iterations called rounds. Finally the pre-output is passed through a final permutation (*FP)* [1-4].
Before the phase of rounds, the input block is divided into two 32-bit halves and processed alternately. This scheme structure ensures that encryption and decryption are very similar processes to each where the only difference is that the subkeys are applied in the reverse order for decrypting process[1-4].

The $\oplus$ symbol denotes XOR operation and the F-function scrambles half a block together

with some of the key. The output from the F-function is then XORed with the other half of

the block, and then the two half blocks are swapped before the next round. Unless after the final round, they are not swapped[1-4].

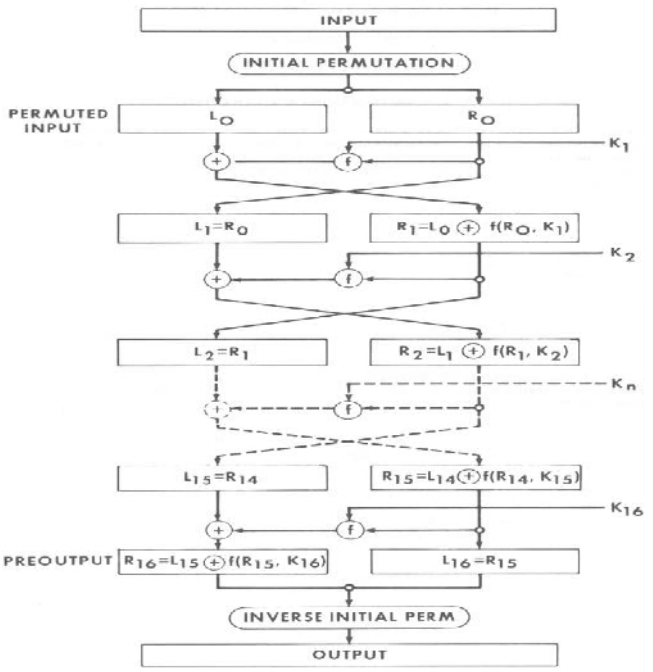


Figure (1): *General depiction of DES Encryption algorithm*

The Feistel (F) function

The F-function, depicted in Figure 2, it consists of four sequential steps are

1. ***Expansion*** :- the 32-bit half-block is expanded to 48 bits using the expansion permutation, denoted *E* in the diagram.
2. ***Key mixing***:- the result is XORed with *a* subkey. Sixteen 48-bit subkeys (one for each round) are derived from the main key using the key schedule.
3. ***Substitution***:- after mixing with the subkey, the block is divided into eight 6-bit pieces. Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a lookup table. The S-boxes provide the core of the security of DES without them, the cipher would be linear, and trivially breakable.
4. ***Permutation***:- finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the P-box [1-4].

The alternation of substitution from the S-boxes, and permutation of bits from the P-box and

E-expansion provides so-called "confusion and diffusion" respectively [1-3].

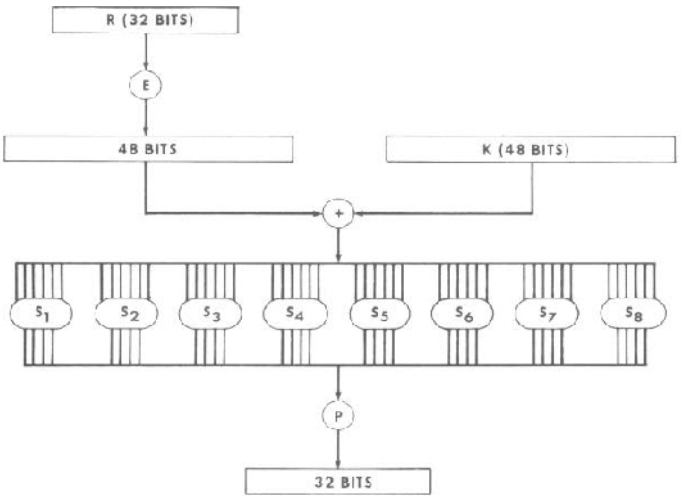


Figure (2): *The Feistel function of DES*

Key schedule

Figure 3 illustrates the key schedule of DES. Initially, 56 bit of the key are selected from the initial 64 bits block by a first permutation. The 56 bits are then divided into two 28-bit halves, each half is thereafter treated separately. In successive round, both halves are rotated left by one and two bits (specified for each round), and then 48 subkey bits are selected by another permutation choice[1-4].

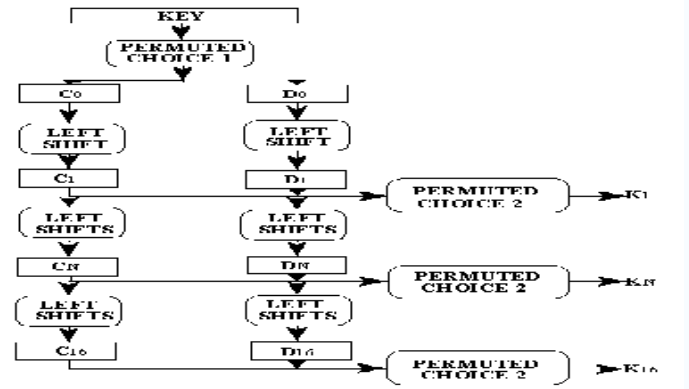


Figure (3): *key schedule of DES*

<u>3- Cryptanalysis of the DES:</u>

Now we will give an over view for the cryptanalysis techniques used to break the DES.

a- Exhaustive search

Exhaustive search is the most basic method of cryptanalysis, also known as the *brute force* attack. This method uses merely one block of known plaintext and the resultant ciphertext to find the secret key. With this pair of known plaintext/ciphertext, it will take 2^{56} maximum DES calculations to find the correct secret key in DES encryption. With today's processing power this not only is possible; indeed it has been performed time and time again. Although exhaustive search is related entirely to the DES secret key length of 56 bits rather than s-boxes, this information is valuable as a baseline for the usefulness of other cryptanalysis techniques [7].

b- Differential cryptanalysis:

A method which analyses the effect of particular differences in plaintext pairs on the differences of the resultant ciphertext pairs. These differences can be used to assign probabilities to the possible keys and to locate the most probable key. The idea behind differential cryptanalysis is to throw out key choices that are unlikely, and keep choices that are very likely. From this reduced subset, a cryptanalyst can run an exhaustive search to find the correct key. In order for differential cryptanalysis to be successful, 2^{47} chosen plaintext/ciphertext pairs are needed. The processing overhead also is less than exhaustive search at 2^{47}. However, Schneider suggests that "the enormous time and data requirements to mount a differential cryptanalytic attack put it beyond the reach of almost everyone."[6].

c- Linear cryptanalysis:

This method attempts to find a linear relation among the plaintext, ciphertext, and keys as they pass through the s-boxes. With enough known plaintext/ciphertext pairs as data, a relation with a high enough probability can be used to find the key. Matsui generated linear approximation tables for the 8 DES s-boxes and found the strongest linearity in S5 (the fifth s-box). The tables were created by analyzing all the combinations of the input and output bits of the s-boxes. Since there are 6 input bits and 4 output bits, there are 1024 ($2^6 \cdot 2^4$) entries in his tables for every s-box. A linear approximation is stronger if it is significantly greater or less than 50% probability. That particular entry in S5 had a value of -20, representing a probability of 12/64 (*1/2 - 20/64*). This value is considered strong enough, and it allows the linear cryptanalysis on DES to be possible. In order to achieve approximately an 85% success rate using this attack method, 2^{43} known plaintext/ciphertext pairs are needed. However, processing overhead is less than the exhaustive search method at 2^{43}. This will help to define the restrictions on s-boxes to make them more resistant to linear

cryptanalysis. He found that increasing the number of output bits of an s-box can endanger the s-box significantly to linear cryptanalysis [5].

4.The proposed DES algorithm:

Two approaches are introduced for modification of DES. The first approach makes the values of S-box key dependent not fixed as in normal DES algorithm. In the second approach the S-box values are varies according to the key and every input data block (key and plaintext dependent).

4.1. The first approach:

The brute force attack was applied to the DES algorithm due to its limited key size. Also the differential and linear cryptanalysis were succeeded due to the static S-boxes construction of the DES [1-6]. Hence the idea was grown to increase the key length and adding a dynamic sense to the S-boxes construction, This means that the S-boxes construction will be a function of the input key, this approach is preliminary supposed to be more complex and exhaustive for cryptanalysts.

The 64 bits key is increased to 152 bits (Figure 4.a), these extra bits used as follow. The first 64 bits from the key used as same as the key in the standard DES, then the next 24 bits used to select the ordering of S-boxes that used in DES (Figure 4.b), then the last 64 bits used to select the ordering of the rows inside the S-boxes for each one individually (Figure 4.c).
These modifications make the S-boxes are dynamically depending on the input key as illustrated above. Also, the extended key size provides a security enhancement against cryptographic attacks which make DES be approvable secure.

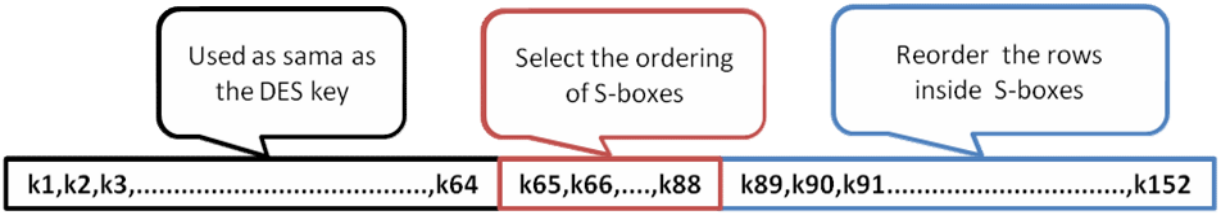
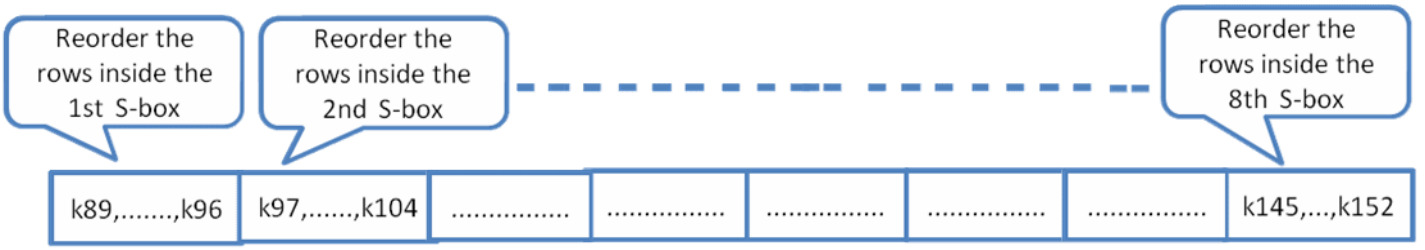


Figure (4.a): *the extended key*

Figure (4.b): *S-boxes reordering*



Figure(4.C): *The rows reordering*

4.2. The second approach:

On the other side, We have introduced another approach in which the S-boxes varies as a function of the input key for every input data block within the same session key . This will provide a misleading and inefficient amount of information needed by the cryptanalysts to perform different kinds of attacks, also this will solve the problem of repeated ciphertext blocks for repeated input blocks for electronic code book mode of operation where, when the input blocks of data are repeated, the S-boxes are varying for every input data block and hence the corresponding cipher texts will not repeated. The idea is illustrated by the following example:

Assume that the key in hexadecimal is :

67784c44d8393f393639a392b7f363b31bb156

and the repeated plaintext is :

standard standard standard standard standard

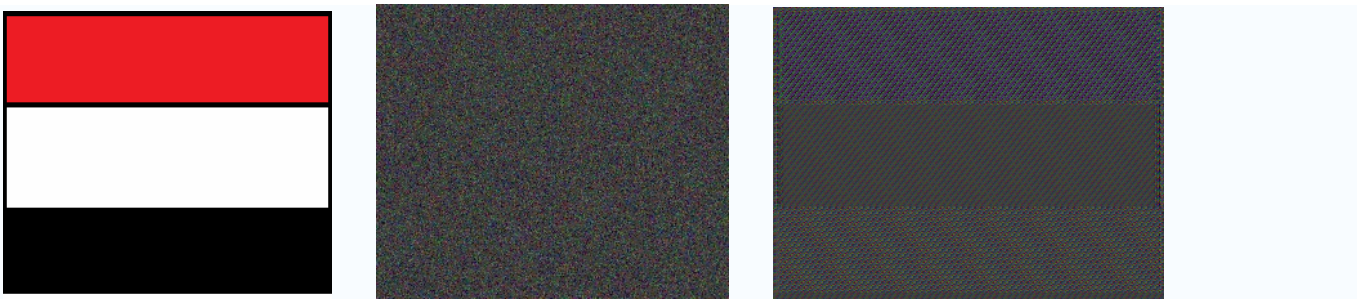
Then the DES ciphertext output will be:

6®Z qY 6®Z qY 6®Z qY 6®Z qY 6®Z qY

And the output ciphertext for the proposed DES will be :

Ý N<¦ Øú,A†5ó ‡&–jLA}þéj ÍËó ðÍq,ÔÿÜ¾

This effect of the proposed DES becomes more obvious when we use it to encrypt pictures as shown in Figure 5.



Picture to be encrypted Proposed DES Standard DES

Figure (5): *Encryption of picture using DES and the proposed DES*

DES is most vulnerable to differential, linear cryptanalysis attacks due to the static behavior of S-boxes, but this modification makes the S-boxes behavior is dynamic (unpredictable) depending on the input key, that will make these cryptanalysis more difficult to attack.

5. Performance Evaluation and Simulation results:

Several performance metrics are considered for performance evaluation, Avalanche effect, security strength and encryption/decryption time. These metrics are tested for the proposed two approaches and the standard DES algorithm.

5.1 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 or one bit of the key should produce a change in many bits

of the cipher text. If the change were small, this might provide a way to reduce the size of the plaintext or key space to be searched [1]. DES exhibits a strong strict key avalanche criteria (SKAC) and plain text avalanche criteria (SPAC) [1], these criteria are tested for the proposed DES algorithm. We found that when one bit changes in the key (whither that bit in the first 64 bits or in any position in the extra added bits) the output ciphertext bits will changed with percentage about 50%. Now we will give an example to show the SKAC for the proposed DES if we change the last bit in the key (the new added bits) for the previous example and then calculate the SKAC .

The key in hexadecimal is :

67784c44d8393f393639a392b7f363b31bb157

Then the output cipher text will be:

y`qÃ~pÙ+d ¯k†± Îä'’5Ü9 FÜ▪¶½ ô9±Á '®;_x

Now, we convert the two output cipher texts from text to binary (ASCII Code) and calculate the SKAC, We find that the length of each cipher texts is 320 bits and 161 bit positions are different which are 50.3125% of the total length this means, about half of the cipher text bits will change if we changed one bit in the input key that satisfies SKAC.

5.2 The strength of the algorithm:

An assessment of the strength of the algorithm, based on key length, algorithm complexity and the best methods of attack. The security of a symmetric cryptosystem is a function of the length of the key. The longer the key, the more resistant the algorithm is to a successful brute force attack. For this reason, key length was chosen as the first parameter for specifying the strength of the cryptographic algorithms, Also for this reason we thought in increasing the key length of DES. The differential and linear cryptanalysis were succeeded due to the static S-boxes construction of the DES, For this reason, We thought in providing dynamic S-boxes configuration depending on the input key that improve the weak round function against cryptographic attacks and make the algorithm more complex.

A brute-force attack involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. We made a program with C Sharp programming language to calculate the average time required to break the encryption algorithms using the brute force attack. If we apply the brute force attack to the proposed DES and compare the time required to break with the time required for the standard DES, We used 5 GHz processor speed and 1 million Computer sets working in parallel for our calculations, We found that, the DES takes

7.2058 sec and the proposed DES takes 1.07E+19 years.

In the next section the Encryption/Decryption time of the proposed DES compared with the standard DES and the simulation results are introduced.

5.3 Encryption/Decryption Time and Simulation results:

In evaluation, we used Pentium IV 1.6 GHz CPU with file with different sizes (41 Kbytes to 7.590Mbytes). The encryption time is considered the time that an encryption algorithm takes to produce a cipher text from a plaintext [8]. This time is used to determine the throughput of the encryption scheme. It indicates the speed of encryption algorithm. The throughput of the encryption scheme is defined as the total plaintext in bytes encrypted divided by the encryption time [9]. As the throughput value is increased, the power consumption of this encryption technique is decreased and vice versa. A comparison is conducted between the results of the proposed algorithm with the standard DES encryption and decryption algorithm in terms of the encryption time and a study is performed on the effect of changing packet size at power consumption during throughput for each DES cryptography algorithm.

The simulation results for this compassion are given in Table 1 and Figures 6, 7 .The results show the superiority of DES algorithm over the proposed DES in terms of the processing time, this difference is just noticeable because the difference in encryption time come from the time needed to establish the S-boxes once at the beginning of the encryption process. Finally, it is found that the second approach has a low performance and low throughput compared with other standard DES an approach number one. This is due to the time needed to establish the S-boxes at the beginning of the encryption process in the first approach is needed for every input data block to be encrypted, but the gain is the complexity of the algorithm and solving the problem of repeated data of electronic code book mode of operation.
These results show that the power consumption for the second approach is the greatest one because it has extra operations that it needs to construct the S-boxes for every input data block. The results show that there is a trade-off between power consumption and security as in wireless networks [10].

Table (1): *Comparative execution times (in seconds) of encryption algorithms with different packet size*

Input size in (Kbytes)	**DES**	**Proposed DES** *(1st approach)*	**Proposed DES** **(***2nd approach* **)**
41	1.521	1.608	1.902
51	1.796	1.839	2.312
100	3.526	3.761	4.341
230	8.122	8.160	10.539
338	11.950	12.033	14.608
513	17.984	18.121	22.170
1060	38.371	38.624	47.151
2480	89.367	90.058	109.184
5110	183.739	185.293	231.044
7590	274.058	274.753	343.952
Average Time	63.0434	63.425	78.7207
Throughput (k bytes/sec)	*27.77926*	*27.61214*	*22.24712*

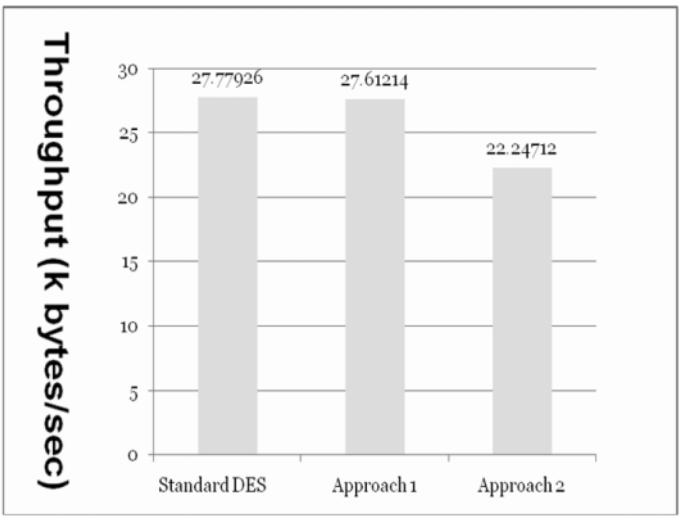


Figure (6): *Throughput of each encryption algorithm (Megabyte/Sec)*

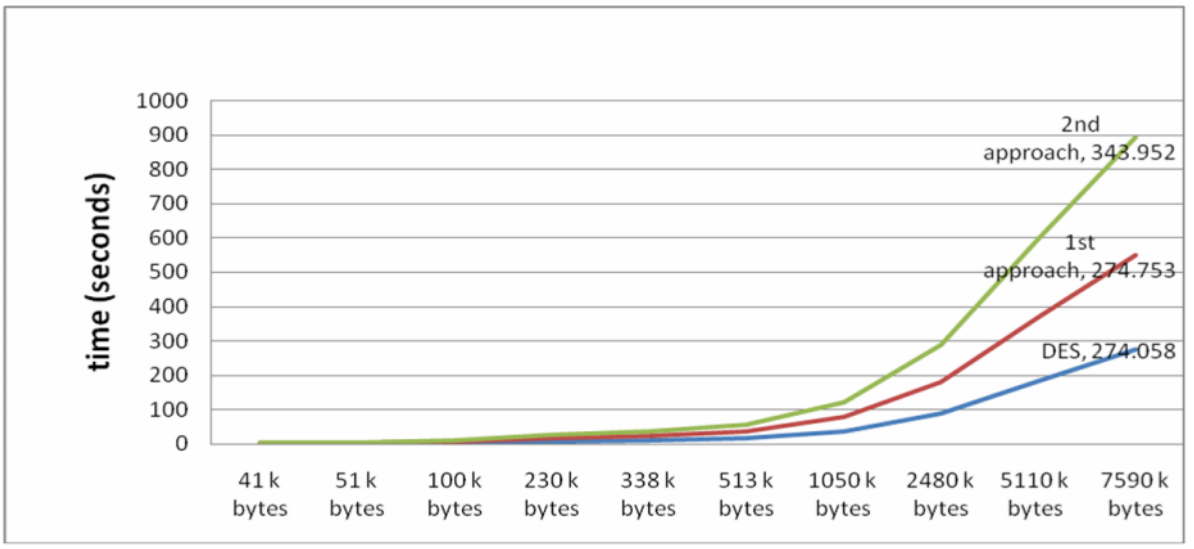


Figure (7): *Time of encryption at different data sizes*

***6. Conclusions*:**

This paper presents two approaches for modification in DES encryption algorithm by providing dynamic S-boxes configuration depending on the input key and input data. These approaches ensure security enhancement by increasing the key length to 152 bits and improve the weak round function against cryptographic attacks.

The results present the dependence of the S-boxes upon the key and the input data blocks and how this complicate the encryption process and improve the algorithm immunity against cryptanalysis attacks. Also, the results illustrate how these approaches solve the problem of the repeated plain text blocks when the algorithm operates at the electronic code book mode of operation. The expense of this is that the throughput of these approaches will be lower than the standard DES.

It is obvious that these approaches could be applicable to any symmetric block cipher algorithm such as Advanced Encryption Standard (AES). This will lead to a great added security and immunity to cryptanalysis attacks.

References:

[1] W.Stallings, "Cryptography and Network Security," Prentice Hall , 2005.

[2] Coppersmith, D. "The Data Encryption Standard (DES) and Its Strength Against Attacks." I BM Journal of Research and Development, May 1994, pp.243-250.
[3] J. Grabbe, "The DES Algorithm Illustrated,"<http://www.aci.net/kalliste/des.htm> (12 July 2002).
[4]"FIPS PUB 46-2," Federal Information Processing Standards Publication (1993), <http://www.itl.nist.gov/fipspubs/fip46-2.htm> (14 July 2002).
[5] Mitsuru Matsui, "Linear Cryptanalysis Method for DES Cipher," EUROCRYPT'93.
[6] Eli Biham, Adi Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," Technical Report CS90-16 of Weizmann Institute of Science, 1990.
[7] Antoine Joux,"Algorithmic cryptanalysis," Taylor and Francis Group, 2009.
[8]M.Hadhoud,D.Salama, "Performance Evaluation of Symmetric Encryption Algorithms," International Journal of Computer Science and Network Security, VOL.8 No.12, December 2008.
[9] A.A. Tamimi, "Performance Analysis of Data Encryption Algorithms," Retrieved October 1, 2008 from http://www.cs.wustl.edu/~jain/cse 56706/ftp/encryption perf /index.html.
[10] K. McKay, "Trade-offs Between Energy and Security in Wireless Networks master Thesis," Worcester Polytechnic Institute, April 2005.