

Dynamic extended DES


Yi-Shiung Yeh ^{1,*}

I-Te Chen ^{2,†}

Ting-Yu Huang ^{1,‡}

Chan-Chi Wang ^{1,§} 

¹*Department of Computer Science and Information Engineering
National Chiao-Tung University
1001 Ta-Hsueh Road, HsinChu
Taiwan 30050
R.O.C.*

²*General Education Center
Kaohsiung Medical University
Taiwan 
R.O.C.*

Abstract

The original S-boxes of DES are important algorithms to resist differential attack. Furthermore, Yeh and Hsu proposed the extended DES, which developed eight more new S-boxes with the same cryptographic properties as original S-boxes in DES. These 16 S-boxes are used to construct the extended DES, which double the block cipher and key size. As a result, a time complexity of differential cryptanalysis of the extended DES is 2^{110} . In this paper we propose an intricate extended DES that includes permutation on S-boxes. By keeping the permutation information in secret, the new version of extended DES is stronger to defeat differential and linear attacks 20922789888000 times.

Keywords : S-Boxes, DES, block cipher, differential attack, linear attack.

*E-mail: ysyeh@csi.nctu.edu.tw

†E-mail: itchen@kmu.edu.tw

‡E-mail: tingyu@csi.nctu.edu.tw

§E-mail: wcc@cyu.edu.tw

Journal of Discrete Mathematical Sciences & Cryptography

Vol. (), No. , pp. 1–10

© Taru Publications

1. Introduction

DES, which is encrypting 64 bits of data block with a 56-bit key size, is one of the most popular block ciphers. The small key size and modern increasingly computing power make DES unsafe even under exhaustive search attack. Therefore, a cipher based on DES with larger key size is necessary.

Each S-box consists of four boolean functions from 6-bit input data to 4-bit output data. It is the major part of DES to defend against cryptanalysis. S-boxes are designed to defend against differential attack [1, 2, 3]. In addition to the public known design criteria, many cryptographic properties of S-box have been studied [5, 6, 7, 8].

Differential attack [1, 2, 3] makes use of the exclusive-or difference of plaintext and ciphertext pairs. It estimates the probability that certain plaintext difference will result in a certain ciphertext difference. This exclusive-or difference sequence of plaintext, intermediate state and ciphertext is the characteristic. As a result, a plaintext-ciphertext pair is the right pair for a characteristic if their XOR sequence in encryption is the characteristic. Right pairs could be used to analyze the correct key value, oppositely the analysis of wrong pairs suggest random values. To reduce the complexity of differential attack, one has to find a high probability characteristic.

Biham and Shamir had shown that DES could be broken by 2^{47} chosen plaintexts or 2^{55} known plaintexts differential attack. Many modified variants of DES result in a weaker DES-like cipher [3]. DES with independent sub keys can be broken by 2^{60} chosen plaintexts or 2^{61} known plaintexts differential attack.

To increase key size without loss of strength against differential attack, eight more S-boxes are proposed [4]. They have the same cryptographic properties and design criteria as original S-boxes. These 16 S-boxes are used to construct the extended DES that has a 112-bit key size. We propose an intricate extended DES that includes permutation on S-boxes. By keeping the permutation information in secret, the new version of extended DES is stronger to defeat differential and linear attacks 20922789888000 times. Furthermore, this method also can be used in any other S-boxes.

2. Attacks to DES

DES was published in two decades before. Up to date, many cryptanalysis has been enforced on it. In addition to the brute-force, differential and linear attacks also threaten DES, although the threats are not yet awful to break it.

Differential attack, mainly used to attack block ciphers, is a famous cryptanalysis method introduced by Biham and Shamir in 1990 [1,2,3]. By this method, the cryptanalysts take care the difference of plaintext and ciphertext pairs. They estimate the probability that certain plaintext difference will result in certain ciphertext difference including the intermediate difference pattern in a block cipher with the same key. A difference pattern with high probability will be useful for deduction of some key bits. The knack is by throwing down a plaintext pairs with the required difference through the cipher to get the corresponding ciphertexts, and then some key values will be suggested according to the ciphertexts and the difference pattern. A right guess of the difference pattern will suggest some key values including the right one; oppositely a wrong guess may suggest incorrect key values. For the high probability of appearance of the particular difference pattern, the right key values will be suggested with the highest frequency while enough number of plaintext pairs have been analyzed.

Linear attack is another powerful cryptanalysis technique, which proposed by M. Matsui [9] in 1993. This attack uses linear approximations to describe the action of a block cipher. By analyzing the structure of the block cipher, especially the S-boxes, some plaintext and ciphertext bits are found to bias equal to 0 or 1 while XOR together. This bias can be exploited to guess some key bits by examining masses of plaintext-ciphertext pairs.

Both of the above two attacks are heavily dependent on the structure of S-boxes. DES has public and fixed S-boxes, this favors the adversary to apply the two attacks.

3. The extended DES

The extended DES [4] has exactly the same data flow and concept of DES. The more eight S-boxes are used in the extended DES to double the block cipher and key size. Some modifications are necessary to P-box and key scheduling algorithms.

The extended DES encrypts 128-bit data block by 112 key bits. All data bits go through an initial permutation. The data bits then split into

two 64-bit data blocks that are right and 3 left data blocks. Two data blocks then go through 32 identical rounds, there is no swap of two data blocks in the last round. After the last round, two data blocks are combined into a 128-bit block. The result will be through the inverse initial permutation.

In each round, the right data block and 96-bit sub-key (R_i and K_i in Figure 3.1) are combined by a round function called F . The output of F is then combined with left part data block by XOR operation. The two data blocks swap in the next round.

The 64-bit right data block is expanded to 96 bits by expansion permutation after combining with the 96-bit sub-key; the 96-bit data is distributed to all 16 S-boxes as input. Each S-box has 4 output bits. Therefore, 64-bit data is used in the next step; and where P-box is permutation box.

Eight more new S-boxes are proposed in following tables. Table 3.1 shows the cryptographically similarity of new S-boxes and original S-boxes. And they are also semi-similar. The new S-boxes are listed in Table 3.2.

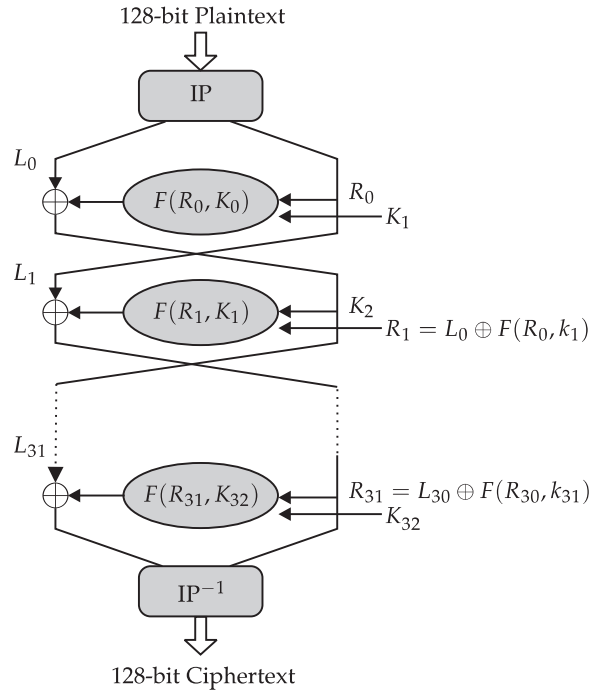


Figure 3.1
128-bit extended DES

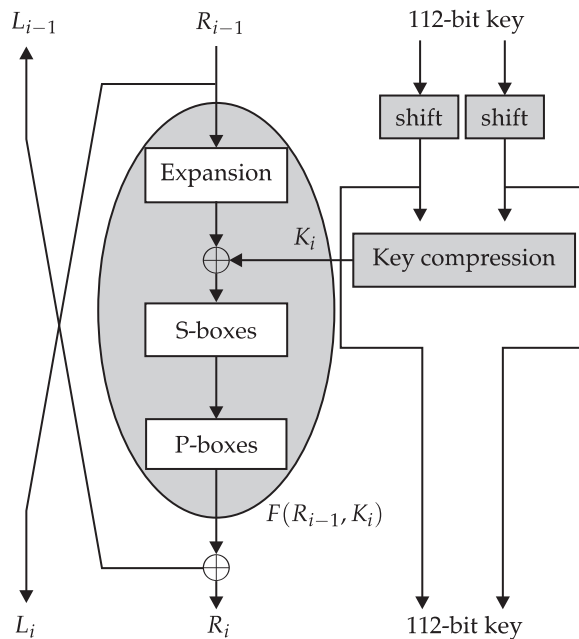


Figure 3.2
One round of 128-bit extended DES

4. Permuted S-boxes

Extended DES has sixteen fixed S-boxes, each of them is a mapping from $\{0, \dots, 63\}$ to $\{0, \dots, 15\}$, or formulary $S : [0 \dots 63] \rightarrow [0 \dots 15]$, used in a settled order. Unfortunately, this usage is convenient for cryptanalysis. To remedy the situation, more complicated use of S-boxes can effectuate.

The change is to rearrange the order of S-boxes in the succeeding round. In detail, a permutation mappings $p : [1 \dots 16] \rightarrow [1 \dots 16]$ is used to construct the new order. The i th S-box in the j th round will be equal to the $p(i)$ th S-box in the $(j - 1)$ th round. For example, the S-boxes sequence in the former round is $S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9 S_{10} S_{11} S_{12} S_{13} S_{14} S_{15} S_{16}$ and given the permutation as $(3, 9, 16, 2, 11, 7, 10, 8, 1, 12, 4, 14, 6, 13, 5, 15)$, then the S-boxes sequence in the next round is $S_3 S_9 S_{16} S_2 S_{11} S_7 S_{10} S_8 S_1 S_{12} S_4 S_{14} S_6 S_5 S_{15}$. By keeping the permutation information in secret, the exact usage of S-boxes is not explicit. This increases the difficulty of cryptanalysis.

Table 3.1
The similarity of new and original S-boxes

New design	Original	Lst	B1	B2	C order	GD	ID	OD	L1	L2	L3	L4	GL	None-zero rate
S-box #9	S-box #1	20	3	3	1	9.31	32.25	46.56	18	20	22	18	78	79.4%
S-box #10	S-box #2	28	3	3	1	11.22	35.81	56.32	22	20	18	18	78	78.6%
S-box #11	S-box #3	24	3	4	1	12.65	41.70	63.62	18	22	20	18	78	79.6%
S-box #12	S-box #4	12*	3	2*	2*	8.16*	32.66	44.00	22	22	22	22	88	68.5%
S-box #13	S-box #5	20	3	2*	1	9.90	35.81	55.32	22	20	18	20	80	76.5%
S-box #14	S-box #6	24	3	3	1	11.31	38.85	59.53	20	20	20	20	80	80.4%
S-box #15	S-box #7	24	3	3	1	12.17	43.45	65.18	18	22	14	20	74	77.2%
S-box #16	S-box #8	20	3	2*	1	10.95	38.71	56.21	22	20	20	22	84	77.1%

LST : Linear structure tolerance
 B1 : First order 0-1 balance tolerance
 B2 : Second order 0-1 balance tolerance
 C order : Maximum order of completeness
 GD : Global SAC-map distance
 ID : Input SAC-map distance
 OD : Output SAC-map distance
 Li : Nonlinearity of output bit i
 GL : Global nonlinearity
 None-zero rate : Percentage of none zero entry in the DDT map



Table 3.2
Extended S-boxes

3 0 9 7 15 12 6 11 14 13 2 1 5 10 8 4	1 10 15 12 8 3 6 5 13 4 0 7 14 9 11 2	S-box #10	
0 3 5 8 9 15 12 6 13 10 11 7 14 4 2 1	4 7 10 0 15 9 1 12 8 14 3 13 5 2 6 11		
15 5 12 2 0 11 9 14 4 3 1 8 10 6 7 13	2 5 4 10 7 12 9 3 11 8 14 1 13 6 0 15		
9 15 0 5 10 6 3 8 2 12 13 11 4 1 14 7	7 0 9 3 4 15 10 6 2 13 5 14 11 8 12 1		
15 4 12 1 5 10 2 13 3 8 6 11 0 7 9 14	10 7 15 12 4 2 1 11 0 13 5 3 9 14 6 8		
6 13 15 2 8 4 5 11 0 7 9 12 3 10 14 1	6 13 12 0 1 7 11 14 3 8 9 15 10 4 5 2		
4 13 15 10 2 1 8 6 14 3 0 5 11 12 7 9	4 1 2 11 15 12 8 6 7 10 14 5 0 9 13 3		
13 3 1 4 11 14 2 8 7 10 12 15 0 5 9 6	1 11 7 14 12 0 2 5 13 6 4 9 3 10 8 15		
4 7 1 12 14 11 8 2 13 10 6 9 0 5 3 15	2 14 15 0 12 11 9 5 4 13 8 3 1 6 7 10		
13 0 2 7 4 14 1 11 3 12 5 10 15 9 8 6	12 5 9 10 7 0 2 15 3 6 14 13 8 11 4 1		
10 1 12 11 9 2 7 14 6 13 15 4 5 8 0 3	12 2 3 14 15 4 10 9 11 1 5 8 6 13 0 7		
7 11 9 4 2 1 14 13 0 6 10 3 12 15 5 8	1 15 12 5 10 9 7 2 6 8 0 14 3 4 13 11		
13 2 4 7 3 12 8 1 0 15 14 9 5 10 11 6	12 2 10 7 1 4 15 8 11 14 0 9 13 3 6 5		
3 8 14 13 9 2 5 11 15 4 0 10 12 7 6 1	2 1 9 4 7 14 12 11 13 8 3 15 10 5 0 6		
2 11 8 13 15 0 4 14 12 5 1 6 10 3 7 9	1 11 15 8 4 13 2 7 14 0 5 6 3 10 9 12		
13 6 1 8 2 11 14 5 10 9 12 3 7 4 0 15	11 13 6 1 8 2 5 14 4 7 10 12 15 9 3 0		
S-box #9		S-box #11	
S-box #13		S-box #14	
S-box #16		S-box #12	
S-box #14		S-box #16	

5. Substitution words access

The whole S-boxes data can be filled into a table that forms as a two dimensions, 16×64 , matrix. Without loss of generality, let the table be $M[1 \dots 16, 1 \dots 64]$ and the initial S-boxes sequence be $S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9 S_{10} S_{11} S_{12} S_{13} S_{14} S_{15} S_{16}$. The k th word (4-bit) of S_i is placed in $M[i, k]$. While applying an S-boxes permutation p , the S-boxes sequence of first encrypting round will be $S_{p(1)} S_{p(2)} S_{p(3)} S_{p(4)} S_{p(5)} S_{p(6)} S_{p(7)} S_{p(8)} S_{p(9)} S_{p(10)} S_{p(11)} S_{p(12)} S_{p(13)} S_{p(14)} S_{p(15)} S_{p(16)}$; that is, the k th word of the i th S-box is placed in $M[p(i), k]$ now. Generally, the S-boxes sequence of the j th round is:

$$S_{p^j(1)} S_{p^j(2)} S_{p^j(3)} S_{p^j(4)} S_{p^j(5)} S_{p^j(6)} S_{p^j(7)} S_{p^j(8)} S_{p^j(9)} S_{p^j(10)} \\ S_{p^j(11)} S_{p^j(12)} S_{p^j(13)} S_{p^j(14)} S_{p^j(15)} S_{p^j(16)},$$

where $p^j(i)$ denotes to execute the mapping p with j times, like this $p(p(\dots p(p(i)) \dots))$. It is obviously that the k th word of the i th S-box of the j th round is placed in $M[p^j(i), k]$.

According to the above derivation, we know that a word in an S-box can still be easily read from the S-boxes table while includes the S-boxes permutation. The increasing calculations are just some mapping operations; and never exceed 16 of nested mapping because of the 16 rounds of extended-DES. Therefore, the new algorithm is considered the 6 same efficient as extended-DES. While decrypting, the same 16 S-boxes sequences in encryption are used but with reverse order. This does not increase the computing time complexity.

6. Permutation materials

The adopted S-boxes permutation should be kept secret. It can be other secret information added to the system independent with key. This will increase the quantity of secret information; system will be more secure in this viewpoint. On the other hand, there is more secret data have to be managed now; this may raise the load for user.

Alternatively, the S-boxes permutation can also be derived from the key. As an example, we can choose the smallest integer A, B which larger than the key value and relatively prime to 16 as the multiplier. The i th value of the permutation function p , will be $p(i) = (A + iB \bmod 16) + 1$.

7. Security analysis

Both differential and linear attacks need know the exact usage of S-boxes. If we can keep the permutation in secret, the adversary will be

difficult to apply the two attacks. The attack may guess the permutation with rarely $\frac{1}{20922789888000}$ probability and then continues the original attack steps, because sixteen S-boxes can derive $16! = 20922789888000$ different permutations. It is computational inefficiency to guess right permutation.

Furthermore, if higher security is required, the permutations used in each round can be different. That is, uses 16 different permutations, maybe all work on the initial S-boxes sequence, and applies them in different rounds. The probability to guess right permutation is about $\frac{1}{20922789888000^{16}} \cong \frac{1}{1.34869 \times 10^{214}}$. To guess the right one from the immense space is computational impossible.

8. Conclusion

This work proposed the method permuting the S-boxes order in the succeeding round; as a result, the usage of S-boxes become more confused. This change can enhance extended DES to resist differential and linear attacks. In addition, this method also can be used in any other S-boxes. However, the permutation information should be kept secret, otherwise the confusion effect no more exists and even favor to the cryptanalysis.

References

- [1] E. Biham and A. Shamir, Differential cryptosystems, in *Proceedings of Advances in Cryptology – Crypto’90*, Springer-Verlag, 1991, pp. 2–21.
- [2] E. Biham and A. Shamir, Differential cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer, in *Proceedings of Advances in Cryptology – Crypto’91*, Springer-Verlag, 1992, pp. 156–171.
- [3] E. Biham and A. Shamir, Differential cryptanalysis of the full 16-Round DES, in *Proceedings of Advances in Cryptology – Crypto’92*, Springer-Verlag, 1993, pp. 487–486.
- [4] Y. S. Yeh and C. H. Hsu, An extended DES, *Journal of Information Science and Engineering*, Vol. 18 (3) (May 2002), pp. 349–365
- [5] J. Seberry and X. M. Zhang, Highly nonlinear 0-1 balanced boolean functions satisfying strict avalanche criterion, in *Proceedings of Advances in Cryptology – AusCrypt’92*, Berlin, Springer-Verlag, 1993, pp. 145–155.
- [6] W. Meier and O. Staffelbach, Nonlinearity criteria for cryptographic functions, in *Proceedings of EuroCrypt’89*, pp. 549–562.

- [7] X. M. Zhang, Y. Zheng and H. Imai, Relating differential distribution tables to other properties of substitution boxes, *Designs Codes and Cryptography*, 2000.
- [8] W. Millan, L. Burnett, G. Carter, A. Clark and E. Dawson, Evolutionary heuristics for finding cryptographically strong S-boxes, *Information and Communication Security*, 2nd International Conference, 1999.
- [9] M. Matsui, Linear cryptanalysis method for DES cipher, in *Proceedings of Advances in Cryptology – EuroCrypt’93*, Springer-Verlag, 1985, pp. 74–87.

Received May, 2005