Differential cryptanalysis on DES cryptosystem up to eight rounds

Vikas Tiwari*, Ajeet Singh and
Appala Naidu Tentu

C.R. Rao Advanced Institute of Mathematics,
Statistics and Computer Science,
University of Hyderabad Campus,
Hyderabad 500046, India
Email: vikas.tiwari2403@gmail.com
Email: ajeetcs@uohyd.ac.in
Email: naidunit@gmail.com
*Corresponding author

Abstract: Differential cryptanalysis is considered as a powerful technique in the field of cryptanalysis, applied to symmetric-key block ciphers. It is a kind of chosen plaintext attack in which the cryptanalyst has some sets of the plaintext and the corresponding ciphertext pairs of his choice. These pairs of the plaintext are related by a constant difference. Generally, it is the analysis of how differences in input information can affect the resultant difference at the output. In this paper, we present step by step implementation of differential cryptanalysis of data encryption standard (DES) up to 8-rounds.

Keywords: block cipher; data encryption standard; DES; differential cryptanalysis; key schedule algorithm; substitution permutation network; SPN.

Reference to this paper should be made as follows: Tiwari, V., Singh, A. and Tentu, A.N. (2019) 'Differential cryptanalysis on DES cryptosystem up to eight rounds', *Int. J. Information Privacy, Security and Integrity*, Vol. 4, No. 1, pp.1–29.

Biographical notes: Vikas Tiwari received his MTech in Artificial Intelligence from University of Hyderabad, India in 2016. Currently, he is working as a Senior Researcher at C.R. Rao AIMSCS, Hyderabad, India. He has published eight research articles in reputed international journals and conferences. His research interests are in the areas of cryptography, cryptanalysis, machine learning and computer networks.

Ajeet Singh obtained his MTech in Computer Science from University of Hyderabad, India in 2016. Currently, he is working as a Senior Researcher in C.R. Rao AIMSCS, Hyderabad, India. He has published eight research articles in reputed international journals and conferences. His research areas of interest include machine learning, cryptography, systems simulation and modelling, rough sets, knowledge discovery, cryptozoology and oneirology.

Appala Naidu Tentu is an Assistant Professor at C.R. Rao AIMSCS, Hyderabad, India. He received his PhD in Computer Science and Engineering from JNTU Hyderabad. He obtained his Master of Technology in Systems

Analysis and Computer Applications from NIT, Suratkal, Karnataka and Master of Science (MSc) from Andhra University. He published more than 20 research papers in reputed international journals and conferences. His research interests are in the areas of cryptography, cryptanalysis and design of security protocols.

1 Introduction

Cryptosystems are generally divided in two types: *symmetric key cryptosystems*, where the same key is used by the sender and the receiver for encryption and decryption respectively. So the key need to be kept private. Hence the symmetric key cryptosystems are also known as private key cryptosystems. The secure distribution of key associated with symmetric key cryptosystems is a challenging task. Data encryption standard (DES) and advanced encryption standards (AES) are examples of symmetric key cryptosystems (Stinson, 2006; Stallings, 2006). Unlike symmetric key cryptosystems, *asymmetric key cryptosystems*, use two keys, called private key and public key. they relay on one key for encryption and the other for decryption. These two keys are different but are related. The RSA algorithm is an example of an asymmetric key cryptosystem.

The differential cryptanalysis was introduced by Biham et al. in 1990. It is one of the seminal work in the area of cryptanalysis. It is a chosen plaintext attack. In differential cryptanalysis, the main task is to study the propagation of differences from round to round inside the cipher and find specific differences, which propagate with relatively high probability. Such pairs of input-output differences can be used to recover some bits of the secret key (Feistel, 1973).

2 Related work

Firstly, IBM designed an iterated cryptosystem called Lucifer (Feistel, 1973), to overcome the increasing information security need for the data in its products. The complete design and structure of the data encryption standard is given in National Bureau of Standards (1977). The procedure of formal coding, where formal expression of each bit in the ciphertext is the XOR form of the bits of the plaintext and the key was presented in Hellman et al. (1976). The manipulations in a formal way of these expressions may reduce the key search attempt. Schaumuller-Bichl (1981, 1982) explored this method and formulated that it needs a significant amount of system memory, which makes the idea impractical. In 1987, Davies gave a known plaintext cryptanalytic attack on DES. Over the past years, several cryptosystems which are variations of DES were presented. Schaumuller-Bichl (1981, 1983) proposed three such types of cryptosystems. Another such variation is the fast encryption algorithm (FEAL). It was designed to be more efficient and implementable on an 8-bit microprocessor. It’s first version had four rounds (Shimizu and Miyaguchi, 1987a) and it was broken by Den Boer (1988) using a chosen plaintext attack. The inventors of FEAL then gave a new version, called FEAL-8, with 8-rounds (Shimizu and Miyaguchi, 1987b; Miyaguchi et al., 1988). Tiwari et al. (2017) given the attack on DES on 3 and 6-rounds.

2.1 Motivation and contribution

Block cipher is a procedure for encrypting plaintext where key and algorithm are applied to a data block. An example of such a symmetric key cryptosystem is DES. Originally in 1970s, it was developed by IBM. Later many researchers have performed cryptanalysis on DES up to specific rounds. Even though the theoretical cryptanalysis exist in literature, but the practicality of these scenarios is not available up to more number of rounds on DES. The key contributions in this paper are as:

- in our experimentation in this paper, we have performed our cryptanalysis on reduced 3-round, 4-round, 5-round as well as 6-round DES and we could be able to retrieve the correct key
- we further extended our attack procedure and later we performed cryptanalysis on reduced 7-round and 8-round DES.

2.2 Organisation of the paper

Rest of the paper is organised as follows: Section 3 discusses some required preliminaries. In Section 4, we have discussed differential cryptanalysis on 3 round DES. In Section 5, attack on 4-round DES is given. Subsequently, for 5-round attack on DES is discussed in Section 6. Section 7 discusses about the attack on 6-round DES. Cryptanalysis for 7 and 8-rounds is given in Section 8. Finally, Section 9, concludes the paper.

3 Preliminaries

3.1 Description of data encryption standard

DES is based on the Feistel structure. DES has 16 rounds (National Bureau of Standards, 1977). Its structure is shown in Figure 1.

We have 64 bit plaintext block and a 64 bit key as input to the encryption algorithm. The 64 bit plaintext block goes through initial permutation (IP) and this permuted output passes through all 16 rounds. The output of the 16th round passes through 32 bit swap and then to the inverse initial permutation to give 64 bit ciphertext block. The 64 bit key is mapped into 56 bit using permuted choice 1 (PC-1). Then in each round, different 48 bit subkey K_i is given after passing it through left circular shift and permuted choice 2 (PC-2).

3.1.1 Round function

The internal structure of round function is same in all rounds as it is based on feistel structure. It is shown in Figure 2.

Figure 1 DES encryption

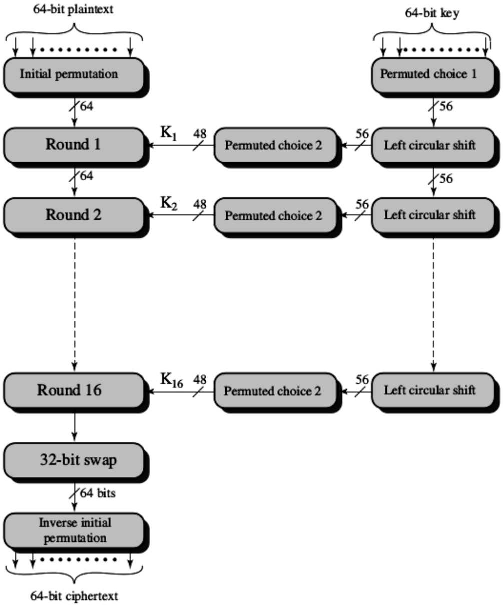
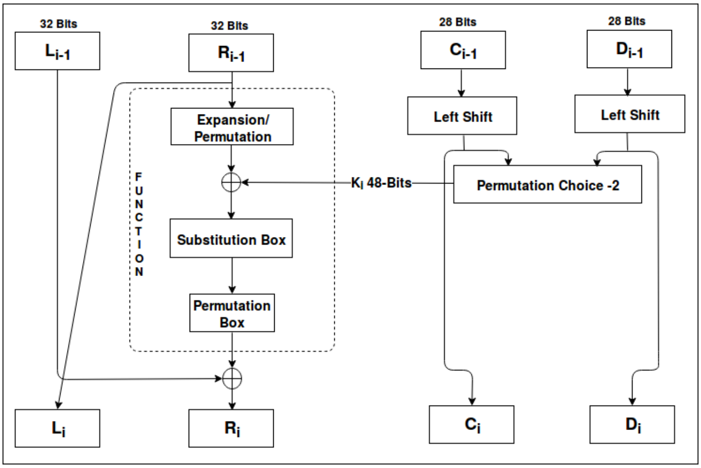


Figure 2 Round function structure



After IP, the plaintext is divided into L and R. This R becomes new L of the next round and new R is the x-or of the previous round L and output of the F function. The whole process can be explained by the following equations:

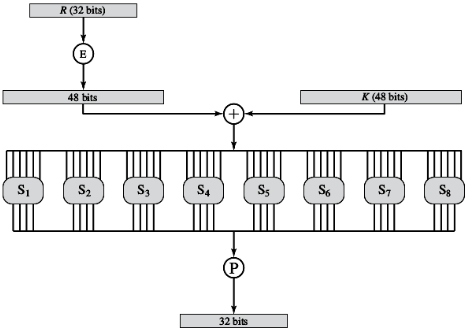
$$L_i = R_{i-1} \tag{1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \tag{2}$$

The 32 bit R_{i-1} is expanded to 48 bits using the expansion table. This 48 bit output is xor-ed with key K_i of 48 bits. Now, this resulting 48 bits is passed through s-boxes to get 32 bits which are further permuted and xor-ed with L_{i-1} to get R_i.

The functioning of s-box is shown in Figure 3.

Figure 3 S-box details



The expanded 48 bit output is xor-ed with 48 bit key and given as input to the eight s-boxes each of which takes 6 bit as input and gives 4 bit output. So this 48 bit xor-ed output is divided into eight blocks of 6 bit each and each of them is given as input to the s-boxes. These 48 bits get transformed into 32 bits as four bit output is obtained from each s-box.

Now, this 32 bit output is permuted and xor-ed with L to get new R.

3.1.2 Key generation

From Figures 2 and 3, we can observe how key is used in the algorithm. The 64-bit key goes through PC-1 where every eighth bit of the key is ignored and converted to 56-bit key. Now this 56 bit key is partitioned into two halves C and D of 28 bit each. In each round these two halves goes through circular left shift of 1 or 2 bits depending on the round separately. These shifted halves are given as input to the next round. For the current round these shifted halves are fed to the PC-2 to produce 48 bit key which is used in $F(R_{i-1}, K_i)$ function.

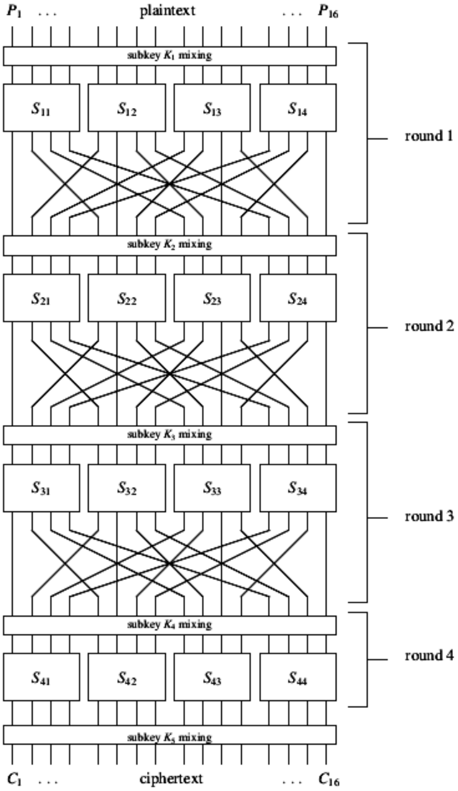
3.2 Substitution permutation network

It is a mechanism similar to Fiestel network that is used to design a block cipher. Here substitution does confusion and permutation does diffusion (Feistel, 1973; Heys and Tavares, 1996).

Confusion is described as being "the use of enciphering transformations that complicate the determination of how the statistics of the ciphertext depend on the statistics of the plaintext". This is achieved by using a complex substitution algorithm. While *diffusion* dissipates the statistical structure of the plaintext within the ciphertext so that attacker cannot determine plaintext corresponding to the ciphertext.

The principle of diffusion and confusion is achieved by applying substitution and permutation to the plaintext over and over again. Iterated block cipher is based on this principle. Thus substitution permutation network (SPN) is a type of iterated block cipher. A basic SPN structure is shown in Figure 4.

Figure 4 Basic SPN structure



It has four rounds. Each round consists of substitution, permutation and key mixing. The input size of plaintext is 16 bit and the key size is 32 bit. Firstly we convert our 32-bit key into round keys of 16-bit each with help of key scheduling algorithm. Now each of these round key is xor-ed with the input it gets in every round. SPN cipher takes 16-bit block of plaintext as input and divides it into four sub-blocks of 4-bits each. Now each of these sub-block goes into key mixing block. After key mixing, it goes to the S-box. The input and output mapping of s-boxes is shown in Table 1. The fundamental property of an S-box is that it is a nonlinear mapping, that is, the output bits can not be represented as a linear function of the input bits.

Table 1 Substitution box

I/P	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
O/P	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Output of s-box is permuted as per the relation given in Table 2. P-box performs the permutation of the bit position.

Table 2 Permutation box

I/P	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
O/P	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

4 Attack on 3-round DES

Though DES has 16 rounds but for cryptanalysis we consider reduced DES with 'n' rounds where n = 3. For this attack we have neglected initial permutation (IP) and its inverse as they do not have effect on cryptanalysis (Stinson, 1995; Biham and Shamir, 1990). To attack 3-round DES, suppose we have a plaintext pair L_0R_0 and $L_0^*R_0^*$ and corresponding ciphertext pair L_3R_3 and $L_3^*R_3^*$. The plaintext pair L_0R_0 and $L_0^*R_0^*$ are chosen so that $R_0 = R_0^*$, 3-round DES structure is shown in Figure 5.

From this figure we can express R_3 as:

$$R_3 = L_2 \oplus f(R_2, K_3) \tag{3}$$

Since L_2 and R_1 are equal,

$$R_3 = R_1 \oplus f(R_2, K_3) \tag{4}$$

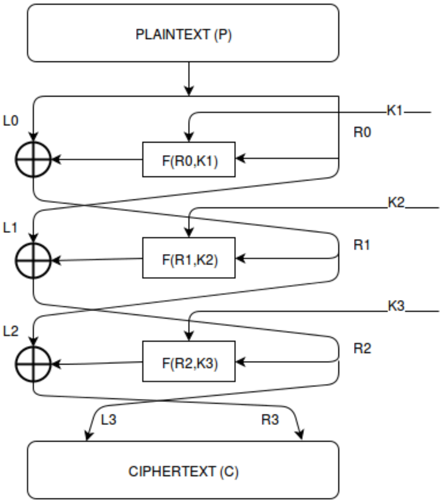
Further R_1 can be expressed as:

$$R_3 = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3) \tag{5}$$

On giving $L_0^*R_0^*$ as input to Figure 5, R_3^* can be expressed as,

$$R_3^* = L_0^* \oplus f(R_0^*, K_1) \oplus f(R_2^*, K_3) \tag{6}$$

Figure 5 3-round DES structure



R'_3 is the xor-ed difference of R_3 and R^*_3. So, $R'_3 = R_3 \oplus R^*_3$.

$$R'_3 = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3) \oplus L^*_0 \oplus f(R^*_0, K_1) \oplus f(R^*_2, K_3) \tag{7}$$

As $L_0 \oplus L_0 * = L'_0$,

$$R'_3 = L'_0 \oplus f(R_0, K_1) \oplus f(R^*_0, K_1) \oplus f(R_2, K_3) \oplus f(R^*_2, K_3) \tag{8}$$

Since $R_0 = R^*_0$ we get,

$$R'_3 = L'_0 \oplus f(R_2, K_3) \oplus f(R^*_2, K_3) \tag{9}$$

We know R'_3 and L'_0 so we can rewrite above equation as,

$$R'_3 \oplus L'_0 = f(R_2, K_3) \oplus f(R^*_2, K_3) \tag{10}$$

Let H and H^* be the two outputs of the eight s-boxes then,

$$f(R_2, K_3) = P(H) \text{ and } f(R^*_2, K_3) = P(H^*)$$

where P is the permutation function. Then,

$$P(H) \oplus P(H^*) = f(R_2, K_3) \oplus f(R^*_2, K_3)$$

$$H' = H \oplus H^* = P^{-1}(R'_3 \oplus L'_0) \tag{11}$$

Now, $R_2 = L_3$ and $R^*_2 = L^*_3$ are also known so we can compute,

$$G = E(L_3) \tag{12}$$

and

$$G^* = E(L_3^*) \tag{13}$$

using the expansion function E. G and G^* are the input to the s-boxes in the third round. The attack on 3-round DES using the triplet G, G^* and H' is as follows.

Let the plaintext pairs and the corresponding ciphertext pairs be:

L_0R_0 : '37580B1359ACEE20'
L_3R_3 : '34E9174A5A2CB621'
$L_0^*R_0^*$: '264A020E59ACEE20'
$L_3^*R_3^*$: '023E68A49B1423D6'

From these pairs, find the s-box inputs for round 3 from equations (12) and (13). That is expand L_3 and L_3^* to 48 bits to get:

$G = 000110101001011101010010100010101110101001010100$
$G^* = 000000000100000111111000001101010001010100001000$

We know that input to the s-boxes is $I = G \oplus K$ where K represents the round key. The exclusive or (x-or) of the inputs of the eight s-boxes is:

$$I \oplus I^* = (G \oplus K) \oplus (G^* \oplus K) \tag{14}$$

Thus, $I \oplus I^* = G \oplus G^*$. So from this we can conclude that input x-or does not depend on the key bits K.

$G' = G \oplus G^*$
$G' = 000110101101011010101110101111111111111101011100$

The output of the s-boxes H' is computed using equation (11).

$L_0' = L_0 \oplus L_0^*$
$L_0 = 00110111010110000000101100010011$
$L_0^* = 00100110010010100000001000001110$
$L_0' = 00010001000100100000100100011101$

$R_3' = R_3 \oplus R_3^*$
$R_3 = 01011010001011001011011000100001$
$R_3^* = 10011011000101000010001111010010$
$R_3' = 11000001001110001001010111110011$

$R_3' \oplus L_0' = 11010000001010101001110011101110$

$H' = P^{-1}(R_3' \oplus L_0')$
$= 01011010000011010010111000100111$

Here P^{-1} is inverse permutation which is shown in Table 3.

Table 3 Inverse permutation

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

Now we have G, G^* and H'. For $1 \leq i \leq 8$, every six bits in G' (G'_i) and four bits in H' (H'_i), we will find pairs whose x-or equal is to G'_i and on giving these pairs as input to the s-box S_i their output x-or is equal to H'_i.

Let these pairs be denoted using Pairs(G'_i, H'_i). If we knew G and G^* we could say,

$$G_i \oplus K_i \in Pairs(G'_i, H'_i) \tag{15}$$

From equation (15), we can conclude that to find key value we can x-or the Pairs(G'_i, H'_i) value with G value. Next step is to tabulate these key values in eight counter array J_i. As each K_i is of 6 bits which would mean 0 to 63 in decimal, the array J_i would range from 0 to 63.

Continuing with previous example, we will find Pairs(G'_i, H'_i) using first 6 bits of the G'_1 and first 4 bits of the H'_1,

$$\text{Pairs}(000110, 0101) = \{110010, 110100\}$$

Here G_1 = 000110, using equation (15),

$$K_1 \in G_1 \oplus \text{Pairs}(000110, 0101) = \{110100, 110010\}$$

Thus we will increment values 52(110100) and 50(110010) in the counter array J_1.

For next pair G'_2 and H'_2, the values will be incremented in the counter array J_2 and so on. We will repeat this process for all pairs in G' and C'. This whole method will be performed with more plaintext-ciphertext pairs until we get a unique value in each of the eight counter arrays J. The position of these unique values determine the key bits.

To get the initial 64-bit key we have to perform few more computations on the result obtained from these eight counter arrays. This is experimentally done and results are attached below.

We have taken three plaintext-ciphertext pairs and computed their G, G^*, G' and H' as explained above. Then found the Pairs(G'_i, H'_i) for $1 \leq \text{i} \leq 8$ and finally J values to be incremented in the eight counter arrays. These pairs are shown below denoted with L_0R_0, $L_0^*R_0^*$, L_3R_3 and $L_3^*R_3^*$.

L_0R_0 = 748502CD38451097	L_3R_3 = 03C70306D8A09F10
$L_0^*R_0^*$ = 3874756438451097	$L_3^*R_3^*$ = 78560A0960E6D4CB
L_0R_0 = 486911026ACDFF31	L_3R_3 = 45FA285BE5ADC730
$L_0^*R_0^*$ = 375BD31F6ACDFF31	$L_3^*R_3^*$ = 134F7915AC253457
L_0R_0 = 357418DA013FEC86	L_3R_3 = D8A31B2F28BBC5CF
$L_0^*R_0^*$ = 12549847013FEC86	$L_3^*R_3^*$ = 0F317AC2B23CB944

For the first pair, the incremented position numbers are in the eight counter arrays $J_1, J_2, \ldots, J_8$.

Pairs$(0,9) = 0,7,40,47, \quad J_1 = 0,7,40,47$
Pairs$(7,6) = 2,53,12,59, \quad J_2 = 5,50,11,60$
Pairs$(56,5) = 4,54,20,38,21,39,25,43, \quad J_3 = 60,14,44,30,45,31,33,19$
Pairs$(14,13) = 50,39,14,44,18,48, \quad J_4 = 11,41,0,34,28,62$
Pairs$(32,5) = 25,56, \quad J_5 = 57,24$
Pairs$(6,11) = 1,19, \quad J_6 = 7,21$
Pairs$(32,6) = 6,39,13,44, \quad J_7 = 28,7,45,12$
Pairs$(12,7) = 35,61,36,58, \quad J_8 = 47,49,40,54$

The output for the second pair is given below. In the same eight counter arrays we incremented the values according to the output shown below:

Pairs$(40,9) = 7,13,52,62, \quad J_1 = 47,37,28,22$
Pairs$(11,12) = 3,46,8,37,14,35,26,55,30,51,$
$J_2 = 8,37,3,46,5,40,17,60,21,56$
Pairs$(63,9) = 44,58, \quad J_3 = 5,19$
Pairs$(52,12) = 0,42,1,43,10,32,11,33,20,62,21,63,30,52,31,53,$
$J_4 = 52,30,53,31,62,20,63,21,32,10,33,11,42,0,43,1$
Pairs$(5,1) = 17,59,29,55, \quad J_5 = 20,62,24,50$
Pairs$(16,15) = 4,38,23,53, \quad J_6 = 20,54,7,37$
Pairs$(11,5) = 0,41,9,32,12,37,14,39, \quad J_7 = 11,34,2,43,7,46,5,44$
Pairs$(54,6) = 0,0, \quad J_8 = 54,54$

Finally, third pair gave the values for eight counter arrays as given below:

Pairs$(59,13) = 4,62,20,46,26,32, \quad J_1 = 63,5,47,21,33,27$
Pairs$(49,5) = 2,45,11,36,27,52, \quad J_2 = 51,28,58,21,42,5$
Pairs$(20,7) = 7,53,16,34, \quad J_3 = 19,33,4,54$
Pairs$(6,5) = 6,34,12,40 \quad J_4 = 0,36,10,46$
Pairs$(35,13) = 17,29,37,41,55,59, \quad J_5 = 50,62,6,10,20,24$
Pairs$(54,11) = 1,2,5,6,17,18,49,50,53,54,57,58,$
$J_6 = 55,52,51,48,39,36,7,4,3,0,15,12$
Pairs$(37,2) = 3,62,7,58,9,52,26,39,27,38,31,34,$
$J_7 = 38,27,34,31,44,17,63,2,62,3,58,7$
Pairs$(31,11) = 46,53, \quad J_8 = 49,42$

At last, we get our J arrays for three rounds. In these arrays we got the unique values at:

$J_1 : 47, J_2 : 5, J_3 : 19, J_4 : 0, J_5 : 24, J_6 : 7, J_7 : 07, J_8 : 49$

We convert these integer values into binary to get 48 bits. We will use key schedule for round three in DES (Biham and Shamir, 1993; Heys and Tavares, 1996; National Bureau of Standards, 1977) to get 48 bits of the key as shown below in Table 4.

Table 4 Key schedule for round 3

51	27	10	36	25	58	9	33	43	50	60	18
44	11	2	1	49	34	35	42	41	3	59	17
61	4	15	30	13	47	23	6	12	29	62	5
37	28	14	39	54	63	21	53	20	38	31	7

This key schedule is for 56 bits, so the rest of the bits will be unknown. Also, our key is of 64-bits. These extra 8 bits are parity bits which will be added based on odd parity. Since very few bits are unknown, we can apply exhaustive search and then calculate odd parity over them. The complete key (in hexadecimal format) is:

‘1A624C8520DEC46’

5 Attack on 4-round DES

We extend the idea of attacking 3-round DES by using probability characteristic to mount attack on 4-round DES. The structure of 4-round DES is given in Figure 6. Before that we define n-round probability characteristic.

n-round probability characteristic: Let $n \geq 1$ be an integer: an n-round characteristic is a list of the form

$$L'_0, R'_0, L'_1, R'_1, p_1, ...L'_n, R'_n, p_n.$$

which satisfy the following properties:

- $L'_i = R'_{i-1}$ for $1 \leq i \leq n$.
- For $1 \leq i \leq n$, let L_{i-1}, R_{i-1} and L^*_{i-1}, R^*_{i-1} be chosen such that $L_{i-1} \oplus L^*_{i-1} = L'_{i-1}$ and $R_{i-1} \oplus R^*_{i-1} = R'_{i-1}$. Suppose L_i, R_i and L^*_i, R^*_i are computed by applying one round of DES encryption. Then the probability that $L_i \oplus L^*_i = L'_i$ and $R_i \oplus R^*_i = R'_i$ is precisely p_i. (Note that this probability is computed over all possible 8 tuples $C = C_1C_2...C_8$.)

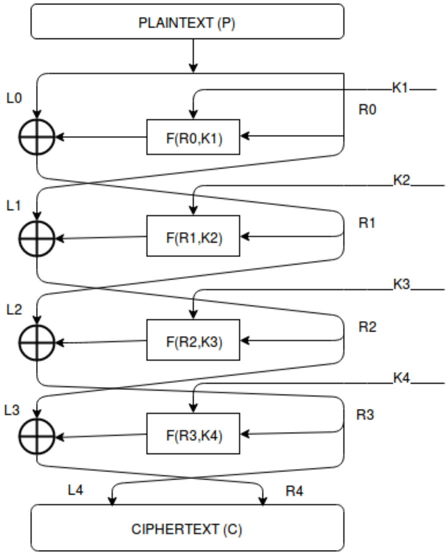
The probability of the characteristic is defined to be the product $p = p_1 \times p_2 \times \ldots p_n$.

To mount an attack on n-round DES we will be using n-3 round characteristic. For suppose, to mount attack on 4-round DES we will use 1-round characteristic, which is shown in Table 5.

Table 5 1-round characteristic

$L'_0 = 40080000$	$R'_0 = 04000000$	
$L'_1 = 04000000$	$R'_1 = 00000000$	$p = 1/4$

Figure 6 4-round DES structure



Even though we can have a probability characteristic with better probability like $p = 1/2$ or $p = 1$, we are not using such characteristic so that we can have $R_1' = 00000000$ which expands to be all zeros. This helps us in getting key bits entering all eight s-boxes. That is we can get 48 bits of key.

Now, we formulate the expression for output xor of s-boxes in fourth round and inputs to the fourth round as we did in previous section.

plaintext pair: $L_0R_0, L_0^*R_0^*$
ciphertext pair: $L_4R_4, L_4^*R_4^*$

We can express R_4 and R_4^* in terms of R_0 and R_1 as follows:

$$R_4 = L_3 \oplus f(R_3, K_4)$$
$$R_4 = R_2 \oplus f(R_3, K_3)$$
$$R_4 = L_1 \oplus f(R_1, K_2) \oplus f(R_3, K_4)$$
$$R_4 = R_0 \oplus f(R_1, K_2) \oplus f(R_3, K_4)$$
$$R_4^* = R_0^* \oplus f(R_1^*, K_2) \oplus f(R_3^*, K_4)$$

So when we xor R_4 and R_4^*

$$R_4' = R_0' \oplus f(R_1, K_2) \oplus f(R_3, K_4) \oplus f(R_1^*, K_2) \oplus f(R_3^*, K_4)$$

From the probability characteristic $R_1' = 00000000$. So $R_1 = R_1^*$. Then $f(R_1, K_2) = f(R_1^*, K_2)$. So $f(R_1, K_2) \oplus f(R_1^*, K_2) = 0$. Hence

$$R_4' = R_0' \oplus f(R_3, K_4) \oplus f(R_3^*, K_4)$$

We can express it as

$$\begin{aligned} &P(y) \oplus P(y^*) = R_4' \oplus R_0' \\ &y' = P^{-1}(R_4' \oplus R_0') \end{aligned}$$

R_4' and R_0' are known from the plaintexts and the corresponding ciphertexts. y' is the output xor of fourth round. We know that $R_3 = L_4$ and $R_3^* = L_4^*$. So we can calculate inputs to the fourth round from corresponding ciphertexts.

$$\begin{aligned} &E = E(R_3) = E(L_4) \\ &E^* = E(R_3^*) = E(L_4^*) \end{aligned}$$

The procedure below shows the steps for mounting probabilistic attack on 4-round DES:

Differential attack on 4-round DES

```
Input: $L_0R_0$ , $L_0^*R_0^*$, $L_4R_4$ and $L_4^*R_4^*$
compute $y' = P^{-1}(R_4' \oplus R_0')$
compute $E = E(L_4)$, $E^* = E(L_4^*)$
for i = 1 to 8 do
    compute $PossibleKeys_i(E_i, E_i^*, y_i')$ and update counter $C_i$
```

In deriving expression for R_4' we have considered that $R_1' = 0..0$. From the differential characteristic this is correct with probability 1/4. That means the key bits we calculate using above procedure are correct with probability 1/4. And with probability 3/4 we may get random garbage values.

Definition: Suppose $L_0 \oplus L_0^* = L_0'$ and $R_0 \oplus R_0^* = R_0'$ where $L_0'R_0'$ is defined input xor according to the characteristic. L_0R_0 and $L_0^*R_0^*$ is called *right pair* if $L_i \oplus L_i^* = L_i'$ and $R_i \oplus R_i^* = R_i'$ for all, $1 \leq i \leq n$ where $L_i'R_i'$ are from defined characteristic. And the pair which do not satisfy these conditions are *wrong pair*.

Now we will define a filtering operation to remove wrong pairs from the input.

Filtering operation: A right pair is a pair of plaintexts which satisfy the probability characteristic. In mounting attack on 4-round DES we have derived expression for R_4' assuming $R_1' = 0..0$. If the pair we are considering for computing possible key bits, is right pair, it should give at least on possible key block for each s-box. This is due to the reason that in right pair encryption process R_{n-1}' gets encrypted to R_n' using key. In case, we are not getting at least one key block for any of the s-boxes then that pair is wrong pair. Giving at least one possible key for each s-box is necessary condition only.

Table 6 Filtering operation example

Plaintext	*Ciphertext*
A198F13F56C9C7C3	0879F9C68C2C5FAA
E190F13F52C9C7C3	40D5D36EEB99DE5F

The inputs to the fourth round function are,

$$E : 000001010000000111111001111111111001111000001100$$
$$E^* : 001000000000101101010101111101010011010110101011100$$

For the above plaintext pairs, the R_0 and R_0^* are.

$$R_0 = 01010110110010011100011111000011$$
$$R_0^* = 01010010110010011100011111000011$$

And $R_0' = R_0 \oplus R_0^*$ is

$$R_0' = 00000100000000000000000000000000$$

From the corresponding ciphertexts R_4 and R_4^* are,

$$R_4 = 10001100000101100010111111010101 0$$
$$R_4^* = 11101011100110011101111001011111$$

Their exor is,

$$R_4' = 01100111101101011000000111110101$$
$$R_4' \oplus R_0' = 01100011101101011000000111110101$$

From the expression we have derived for y',

$$y' = P^{-1}(R_4' \oplus R_0') = 11000110111010001111001011010100$$

Table 7 Possible keys for wrong pair

s-box	E_i	E_i^*	y_i'	*Possible keys for s-box i*
1	000001	001000	1100	13, 15, 4, 6, 29, 31, 20, 22, 58, 51
2	010000	000001	0110	4, 21, 36, 44, 53, 61
3	001111	011010	1110	24, 20, 21, 13, 0, 1, 49, 36
4	110011	101011	1000	32, 56, 8, 12, 6, 30, 16, 20
5	111111	111010	1111	51, 54, 40, 41, 44, 45, 10, 15
6	110011	100110	0010	47, 41, 60, 58, 15, 26
7	111000	101101	1101	43, 62
8	001100	011100	0100	

The s-box inputs and output xors and possible keys generated for each s-box are tabulated in Table 7. The procedure to extract possible key bits for the given E_i, E_i^*, and y_i' in previous sections.

From Table 7, it is evident that possible keys for eighth s-box is empty set. So this pair is wrong pair. Hence it will get discarded by filtering operation. Suppose we have a pair for which $|possible\_keys_i| \geq 1$ for all $1 \leq i \leq 8$ then it will survive the filtering operation. That does not mean it is a right pair. Filtering operation helps in removing wrong pairs.

Here, a bit string of length 6 is the binary representation of integers from 0 to 63, which are indices of the counters.

After finding the index with maximum value in 8 different counters C_1 to C_8, convert them into 6-bit binary string (note that the indices are considered in row by row order). The subkey blocks for each s-box are:

$$[C_1 : 010100]$$
$$[C_2 : 100100]$$
$$[C_3 : 001101]$$
$$[C_4 : 001000]$$
$$[C_5 : 101000]$$
$$[C_6 : 111010]$$
$$[C_7 : 100000]$$
$$[C_8 : 101111]$$
$$K_4 = 010100\ 100100\ 001101\ 001000\ 101000\ 111010\ 100000\ 101111$$

After doing inverse of PC-2 and applying inverse of PC-1 on these 48 bits we get,

00x11x1 0x10001 0100110 1000xx0 010x00x x000110 1110110 0100011

x here means unknown bit. As we have eight unknown bits we have to do exhaustive search for $2^8 = 256$ times. And along with that odd parity bits need to be added for each 7-bit blocks. This gives us complete key. The key is

‘1A624C89520DEC46’

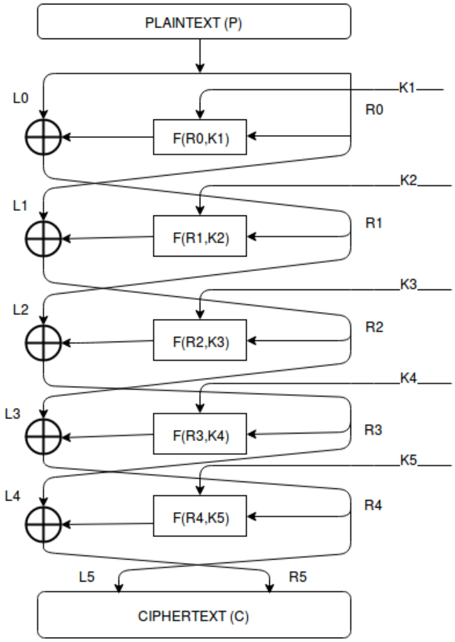
6 Attack on 5-round DES

We use the previously discussed method to mount an attack on 5-round DES. The structure of 5-round DES This will again be a probabilistic attack. So we will use the 2-round probability characteristic, which is shown in Table 8.

Table 8 2-round probability characteristic

$Lh'_0 = 00200008$	$Rh'_0 = 00000400$	
$Lh'_1 = 00000400$	$Rh'_1 = 00000000$	$p = 1/4$
$Lh'_2 = 00000000$	$Rh'_2 = 00000400$	$p = 1$

Figure 7 5-round DES structure



The expression for R'_5 is

$$\begin{aligned}
R_5 &= L_4 \oplus f(R_4, K_5)\\
R_5 &= R_3 \oplus f(R_4, K_5)\\
R_5 &= L_2 \oplus f(R_2, K_3) \oplus f(R_4, K_5)\\
R^*_5 &= L^*_2 \oplus f(R^*_2, K_3) \oplus f(R^*_4, K_5)\\
R'_5 &= L'_2 \oplus f(R_2, K_3) \oplus f(R_4, K_5) \oplus f(R^*_2, K_3) \oplus f(R^*_4, K_5)
\end{aligned}$$

From the probability characteristic $L'_2 = 0$. So the expression for R'_5 will be

$$R'_5 = f(R_2, K_3) \oplus f(R_4, K_5) \oplus f(R^*_2, K_3) \oplus f(R^*_4, K_5)$$

The expansion of R'_2 is , (*Note R'_2 is taken from Table 8, which has 2-round probability characteristic.*)

000000 000000 000000 000000 000000 001000 000000 000000

In the third round except S_6 all the s-boxes are getting zero input difference. So for these s-boxes sub-block of R_2 and R^*_2 are same. The expression gets simplified to

$$R'_5 = f(R_4, K_5) \oplus f(R^*_4, K_5)$$

We denote $f(R_4, K_5)$ and $f(R_4^*, K_5)$ with $P(y)$ and $P(y^*)$ as we did in previous sections.

$$R_5' = P(y) \oplus P(y^*)$$
$$y' = P^{-1}(R_5')$$

The expressions for inputs to the fifth round are:

$$E = E(L_5)$$
$$E^* = E(L_5^*)$$

$R_5',\ L_5\,L_5^*$ are known from the pair of plaintexts and their corresponding ciphertexts. The filtering operation used in 4-round cryptanalysis will be used here also to avoid unnecessary garbage values while counting the key bits. The procedure to mount attack is shown below.

Procedure for mounting attack on 5-round DES

```
Input:
  plaintexts pair: $L_0R_0,\ L_0^*R_0^*$
  corresponding ciphertexts pair: $L_5R_5,\ L_5^*R_5^*$
for each pair
  compute $y' = P^{-1}(Rh_5')$
  compute $E = E(L_5),\ E^* = E(L_5^*)$
  for $i = 1, 2, 3, 4, 5, 7, 8$
    compute possible keys $(E_i,\ E_i^*, y_i)$
    update counter $C_i$
```

After experimenting the sample input consisting of 50 pairs of plaintexts which satisfy the input difference as in the probability characteristic, using the above procedure we will get the maximum values in each counters ranging from C_0 to C_8.

Now, we will take the index with maximum value from each counter and convert it to 6-bit binary string then K_5 including unknown bits is,

000010 001101 000100 010101 111001 XXXXXX 111110 010010

After applying inverse of PC2 on K_5, the key will be:

110101 10x100 00100x 010x00 x0001x 10x1x0 0x1x11 x1x00x 10x10x11

After shifting right circularly by 8 bits (2 + 2 + 2 + 1 + 1 in reverse)

0x00x00011010110x10000100x0110x10x111x10x1x00x1x11x1x00x

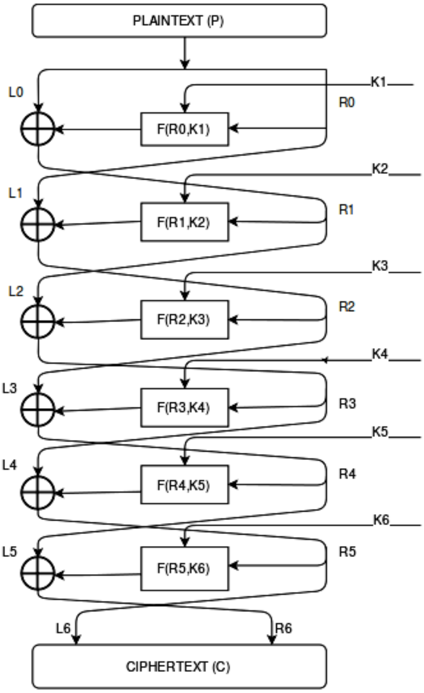
The number of unknown bits is 14. So the exhaustive search space size is $2^{14} = 16384$. The final key after adding odd parity is

‘1A624C89520DEC46’

7 Attack on 6-round DES

We now extend this attack to 6-round DES. To attack 6-round DES, we begin with plaintext pair L_0R_0 and $L_0^*R_0^*$ and their ciphertext pair L_6R_6 and $L_6^*R_6^*$. A 6-round DES structure is shown in Figure 8.

Figure 8 6-round DES structure



We will use 3-round characteristic given in Table 9, to attack 6-round DES.

Table 9 3-round characteristic

L_0' = 40080000	R_0' = 04000000
L_1' = 04000000	R_1' = 00000000
L_2' = 00000000	R_2' = 04000000
L_3' = 04000000	R_3' = 40080000

From Figure 8, we can express R_6 as:

$$R_6 = L_5 \oplus F(R_5, K_6) \tag{16}$$

Since $L_5 = R_4$, we can rewrite above equation

$$R_6 = R_4 \oplus F(R_5, K_6) \tag{17}$$

Since $R_4 = L_3 \oplus f(R_3, K_4)$ we have,

$$R_6 = L_3 \oplus F(R_3, K_4) \oplus F(R_5, K_6) \tag{18}$$

With $L_0^* R_0^*$ as input to Figure 8, R_6^* can be written as,

$$R_6^* = L_3^* \oplus F(R_3^*, K_4) \oplus f(R_5^*, K_6) \tag{19}$$

$R_6' = R_6 \oplus R_6^*$. So,

$$R_6' = L_3 \oplus F(R_3, K_4) \oplus F(R_5, K_6) \oplus L_3^* \oplus F(R_3^*, K_4) \oplus F(R_5^*, K_6) \tag{20}$$

As $L_3 \oplus L_3* = L_3'$,

$$R_6' = L_3' \oplus F(R_3, K_4) \oplus F(R_3^*, K_4) \oplus F(R_5, K_6) \oplus F(R_5^*, K_6) \tag{21}$$

By taking $R_3 = R_3^*$ we get,

$$R_6' = L_3' \oplus F(R_5, K_6) \oplus F(R_5^*, K_6) \tag{22}$$

We know R_6' and L_3' (from Table 9) so we can rewrite above equation as,

$$R_6' \oplus L_3' = F(R_5, K_6) \oplus F(R_5^*, K_6) \tag{23}$$

Let H and H^* be the two outputs of the eight s-boxes in round function. Then,

$$F(R_5, K_6) = P(H) \text{ and } F(R_5^*, K_6) = P(H^*)$$

where P performs the permutation function. Then,

$$P(H) \oplus P(H^*) = F(R_5 K_6) \oplus F(R_5^*, K_6)$$

$$H' = H \oplus H^* = P^{-1}(R_6' \oplus L_3') \tag{24}$$

From Figure 8, $R_5 = L_6$ and $R_5^* = L_6^*$ are known, by this we can compute,

$$G = E(L_6) \tag{25}$$

and

$$G^* = E(L_6^*) \tag{26}$$

using the expansion function E. G and G* are the input to the s-boxes in the sixth round.

From Table 9, we have L'_3 = 04000000 and R'_3 = 40080000. The input x-or to the s-boxes in round 4 is E(R'_3), which is,

001000000000000000010100...0.

where E is the expansion function. As we know input to the each of the eight s-boxes is 6-bit, in this case the input x-or to S2, S5, S6, S7 and S8 is zero and hence their output x-or will also be zero for round 4. This means we can compute output of only these five s-boxes in sixth round.

To attack, we need to compute G, G^* & H' and then find Pairs(G'_i & H'_i) to find values in counter arrays J_2, J_5, J_6, J_7 and J_8 corresponding to s-boxes S2, S5, S6, S7 and S8. With right plaintext-ciphertext pair we will get correct values in J array and wrong pair will give incorrect H' and thus wrong value of Pairs(G'_i & H'_i).

Wrong pair can be detected in the following manner. Suppose we have pair (hexadecimal format).

Plaintext	*Ciphertext*
9468A0BE00166155	3D6A906A6566D0BF
D460A0BE04166155	3BC3B236398379E1

For this pair we will find G, G^* & H' as we did in 3-round attack. Here the input to s-boxes and their outputs are computed as:

Table 10 Sample S-box input and S-box output

J	G_i	G_i^*	H'_i
2	111010	110111	0110
5	110010	110110	0100
6	100000	100100	1111
7	001101	000110	0000
8	010100	101100	1001

Now using $G'\&H'$, we will find Pairs(G'_i & H'_i) and then $J_i \in$ Pairs(G'_i & H'_i) for $i = 2, 5, 6, 7, 8$. This is as follows:

Table 11 Values of counter array J_2, J_5, J_6, J_7, J_8

J	$J_i \in$ Pairs(G'_i, H'_i)
2	
5	
6	4, 8, 12, 16, 41, 45, 52, 56
7	
8	31, 39

From Table 11, we observe that J$_2$,J$_5$ and J$_7$ are empty, so this pair is a wrong pair and will be discarded.

Thus, to survive the filtering operation, a pair should be such that | J$_i$ | $>$ 0 for i = 2, 5, 6, 7, 8.

For 6-round attack we generated plaintext pairs satisfying input x-or L'_0 and R'_0 given in Table 9 and encrypted them with some random but same key to get ciphertext. Then applied filtering operation to get right pairs.

These right pairs give output in counter arrays as shown below:

$$\begin{aligned}
J_2 = {} & 7\ 6\ 10\ 10\ 8\ 8\ 11\ 6\ 6\ 8\ 10\ 11\ 6\ 8\ 9\ 7\ 8\ 10\ 7\ 7\ 6\ 6\ 5\ 10\ 10\ 29\ 9\ 8\ 7\ 6\ 5\\
& 10\ 5\ 8\ 8\ 8\ 7\ 8\ 5\ 7\ 6\ 9\ 7\ 5\ 10\ 9\ 8\ 8\ 8\ 8\ 7\ 8\ 9\ 5\ 10\ 10\ 11\ 7\ 8\ 7\ 8\ 6\ 7\ 9.\\
J_5 = {} & 6\ 5\ 6\ 5\ 7\ 6\ 5\ 4\ 5\ 4\ 4\ 3\ 5\ 6\ 2\ 3\ 5\ 3\ 5\ 5\ 3\ 1\ 2\ 4\ 4\ 5\ 0\ 4\ 3\ 3\ 4\ 4\ 5\ 4\ 6\ 1\\
& 2\ 3\ 6\ 3\ 3\ 1\ 5\ 2\ 2\ 5\ 1\ 3\ 12\ 3\ 2\ 5\ 5\ 9\ 1\ 4\ 6\ 6\ 9\ 2\ 4\ 6\ 3\ 4.\\
J_6 = {} & 5\ 6\ 5\ 5\ 3\ 6\ 7\ 11\ 28\ 9\ 6\ 6\ 6\ 9\ 7\ 9\ 6\ 5\ 7\ 4\ 5\ 2\ 10\ 4\ 5\ 6\ 4\ 6\ 7\ 6\ 10\ 5\ 8\\
& 6\ 5\ 5\ 3\ 2\ 4\ 4\ 6\ 8\ 5\ 10\ 6\ 11\ 5\ 7\ 8\ 8\ 5\ 6\ 6\ 6\ 4\ 7\ 6\ 7\ 6\ 6\ 2\ 6\ 5.\\
J_7 = {} & 7\ 3\ 3\ 3\ 2\ 2\ 4\ 4\ 4\ 3\ 2\ 4\ 6\ 7\ 5\ 7\ 6\ 4\ 7\ 4\ 5\ 7\ 4\ 1\ 1\ 7\ 3\ 3\ 3\ 3\ 4\ 5\ 4\ 5\ 3\ 1\\
& 4\ 5\ 3\ 7\ 3\ 4\ 23\ 3\ 4\ 3\ 5\ 3\ 5\ 3\ 1\ 2\ 3\ 3\ 7\ 6\ 6\ 2\ 3\ 6\ 2\ 4\ 4\ 6.\\
J_8 = {} & 1\ 1\ 5\ 5\ 0\ 1\ 3\ 4\ 1\ 4\ 3\ 2\ 4\ 6\ 4\ 5\ 3\ 4\ 2\ 5\ 8\ 3\ 5\ 5\ 2\ 5\ 2\ 1\ 4\ 5\ 4\ 3\ 7\ 4\ 7\\
& 20\ 2\ 1\ 7\ 4\ 5\ 3\ 6\ 3\ 1\ 2\ 5\ 5\ 3\ 4\ 5\ 0\ 5\ 6\ 2\ 5\ 6\ 3\ 4\ 3\ 2\ 4\ 3\ 3.
\end{aligned}$$

The unique values present in each of these array are at the positions:

J_2: 25 → 011001
J_5: 48 → 110000
J_6: 9 → 001001
J_7: 42 → 101010
J_8: 35 → 100011

This gives us 30 bits out of 48 key bits as done in 3-round attack. We will use another 3-round characteristic given in Table 12 to get more key bits. This will give us 12 more key bits by giving values in J_1 and J_4 array as,

J_1: 55 → 110111
J_4: 18 → 010010

Table 12 Another 3-round characteristic

L'_0 = 00200008	R'_0 = 00000400
L'_1 = 00000400	R'_1 = 00000000
L'_2 = 00000000	R'_2 = 00000400
L'_3 = 00000400	R'_3 = 00200008

Now we have 42 key bits. These key bits are shown below in Table 13. Here ’-1’ denotes unknown key bits as J$_3$ is still not known.

We will apply key schedule of round 6 given in Table 14 on these 48 key bits. The key schedule is of 56 bits as we have seen in 3-round attack.

Table 13 Output of eight counter arrays in 6-round attack

J1:	1	1	0	1	1	1
J2:	0	1	1	0	0	1
J3:	–1	–1	–1	–1	–1	–1
J4:	0	1	0	0	1	0
J5:	1	1	0	0	0	0
J6:	0	0	1	0	0	1
J7:	1	0	1	0	1	0
J8:	1	0	0	0	1	1

Table 14 Key schedule for round 6

3	44	27	17	42	10	26	50	60	2	41	35
25	57	19	18	1	51	52	59	58	49	11	34
13	23	30	45	63	62	38	21	31	12	14	55
20	47	29	54	6	15	4	5	39	53	46	22

Now, we have total of 14 unknown key bits which are computed using exhaustive search. To recover 64-bit key, odd parity will be added after every seventh key bit. The result after applying key scheduling (including parity bit) is shown below in Table 15.

Table 15 Result after applying 6-round key schedule

–1	0	1	1	0	1	–1	–1
–1	1	1	0	1	0	0	–1
1	–1	–1	1	0	1	1	–1
–1	0	0	–1	1	0	1	–1
–1	0	1	–1	–1	0	0	–1
0	1	–1	1	0	1	0	–1
0	1	–1	0	0	0	1	–1
–1	0	1	1	–1	0	0	–1

The final key (in hexadecimal format) obtained is:

‘34E9F71820756231’

8 Attack on 7-round and 8-round DES

In cryptanalysis of 7-round DES we have to use 4-round probability characteristic. The following probability characteristic given in Table 16, is used with differential probability $p = \frac{1}{2621.44}$.

The attack on 7-round DES as shown in Figure 9, we starts with choosing a pair of plaintexts L_0R_0, $L_0^*R_0^*$ and collecting the corresponding ciphertexts L_7R_7, $L_7^*R_7^*$ where $L_0' = 405C0000$ and $R_0' = 04000000$. We write expression for R_7 and expand it so that it can be expressed in terms of $L_4\ and\ R_4$.

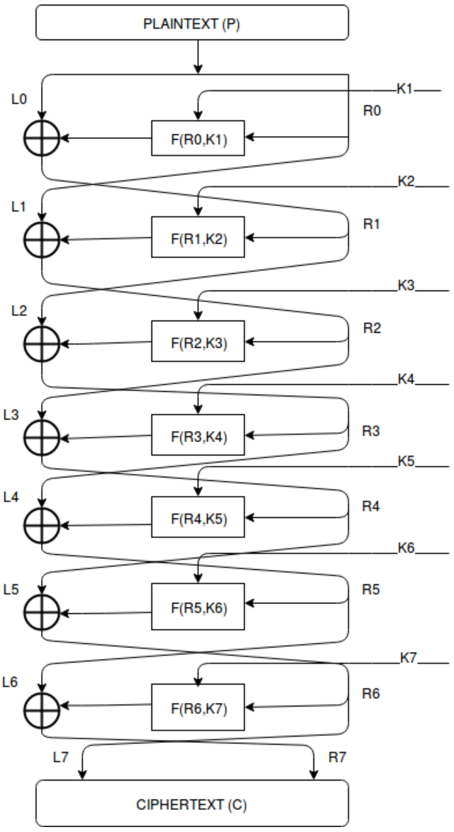
$$R_7 = L_6 \oplus f(R_6, K_7)$$

$$R_7 = R_5 \oplus f(R_6, K_7)$$
$$R_7 = L_4 \oplus f(R_4, K_5) \oplus f(R_6, K_7)$$

Table 16 4-round characteristic

$L'_0 = 405C0000$	$R'_0 = 04000000$	
$L'_1 = 04000000$	$R'_1 = 00540000$	$p = \frac{1}{4}$
$L'_2 = 00540000$	$R'_2 = 00000000$	$p = \frac{10.16}{64.64}$
$L'_3 = 00000000$	$R'_3 = 00540008$	$p = 1$
$L'_4 = 00540000$	$R'_4 = 04000000$	$p = \frac{10.16}{64.64}$

Figure 9 7-round DES structure



Similarly,

$$R_7^* = L_4^* \oplus f(R_4^*, K_5) \oplus f(R_6^*, K_7)$$

And xor of R_7 and R_7^* is,

$$R_7' = L_4' \oplus f(R_4, K_5) \oplus f(R_6, K_7) \oplus f(R_4^*, K_5) \oplus f(R_6^*, K_7)$$

From the probability characteristic L_4' and R_4' are known. R_4' from Table 16 can be expanded using expansion and permutation function to,

000000 001000 000000 000000 000000 000000 000000 000000

From this expansion it is evident that, except S_2 all the S-boxes are getting zero input x-or in the fifth round. So their fifth round output xor will also be zero. So R_7' is,

$$R_7' = L_4' \oplus f(R_6, K_7) \oplus f(R_6^*, K_7)$$

As in previous chapters, we denote $f(R_6, K_7)$ and $f(R_6^*, K_7)$ with $P(y)$ and $P(y^*)$ where P is permutation box in round function and y is output of substitution in round function.

$$P(y) \oplus P(y^*) = R_7' \oplus L_4'$$
$$y' = P^{-1}(R_7' \oplus L_4')$$

This is output xor of substitution in seventh round. And the inputs to the substitution in seventh round are,

$$E = E(L_7)$$
$$E^* = E(L_7^*)$$

Now we use inputs and output xors to calculate possible keys using 7 counters namely $C_1,\ C_2,\ C_3,\ C_4,\ C_5,\ C_6,\ C_7$ as these s-boxes are getting zero input difference in fifth round. This complete procedure can be explained briefly as follows:

Algorithm for 7-round

```
Input:
  Pair of plaintexts: $L_0R_0,\ L_0^*R_0^*$ which satisfy the condition that $L_0' = 405C0000$ and
  $R_0' = 04000000$
  Corresponding ciphertexts: $L_7R_7,\ L_7^*R_7^*$
for each pair of plaintexts
  compute $y' = P^{-1}(R_7' \oplus L_4')$
  $L_4'$ is taken from the probability characteristic.
  compute $E = E(L_7),\ E^* = E(L_7^*)$
  for $i = 1, 3, 4, 5, 6, 7, 8$
    compute possible keys $(E_i,\ E_i^*, y_i)$
    update counter $C_i$
```

With this procedure 42 bits of key can be retrieved. For the remaining key bits one has to do exhaustive search with search space $2^{14} = 16{,}384$.

Attack on 8-round DES

As we already know that for any n-round DES we need probability characteristic with n-3 rounds. So the Table 17 shows 5 round characteristic to be used for 8-round cryptanalysis as shown in Figure 10.

Table 17 5-round characteristic 1

$L'_0 = 405C0000$	$R'_0 = 04000000$	
$L'_1 = 04000000$	$R'_1 = 00540000$	$p = \frac{1}{4}$
$L'_2 = 00540000$	$R'_2 = 00000000$	$p = \frac{10.16}{64.64}$
$L'_3 = 00000000$	$R'_3 = 00540008$	$p = 1$
$L'_4 = 00540000$	$R'_4 = 04000000$	$p = \frac{10.16}{64.64}$
$L'_5 = 04000000$	$R'_5 = 405C0000$	$p = \frac{1}{4}$

The expression for R_8 is,

$$R_8 = L_5 \oplus f(R_5, K_6) \oplus f(R_7, K_8)$$

Similarly R_8^* is,

$$R_8^* =^* \oplus f(R_5^*, K_6) \oplus f(R_7^*, K_8)$$

And R'_8 is,

$$R'_8 = L_5 \oplus f(R_5, K_6) \oplus f(R_7, K_8) \oplus L_5^* \oplus f(R_5^*, K_6) \oplus f(R_7^*, K_8)$$
$$R'_8 = L'_5 \oplus f(R_5, K_6) \oplus f(R_7, K_8) \oplus f(R_5^*, K_6) \oplus f(R_7^*, K_8)$$

From the probability characteristic R'_5 is known. And if we expand and permute it,

001000 000000 001011 111000 000000 000000 000000 000000

So for $S_2,\ S_5,\ S_6,\ S_7,\ S_8$,

$$R'_8 = L'_5 \oplus f(R_7, K_8) \oplus f(R_7^*, K_8)$$
$$R'_8 = L'_5 \oplus P(y) \oplus P(y^*)$$
$$y' = P^{-1}(R'_8 \oplus L'_5)$$

So from this we can calculate 30 key bits. For the remaining key bits we need another probability characteristic which is shown in Table 18.

The expansion of R'_5 is,

000000 001000 000000 001000 000000 001111 110000 000000

This characteristic can result in 18 key bits. For the remaining key bits we need to do exhaustive search. The number of pairs need to be tried for this is around 4 lacs for each probability characteristic. To increase the probability of getting the key we can include some more filtering operations as discussed in previous sections.

Figure 10 8-round DES structure

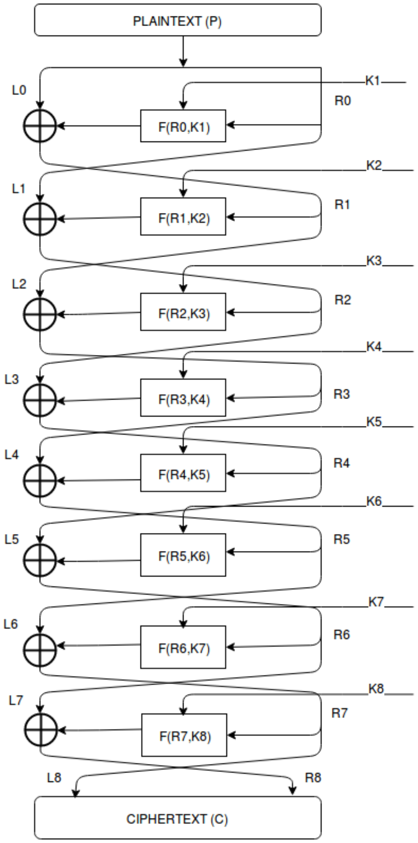


Table 18 5-round characteristic 2

$L_0' = 04040780$	$R_0' = 00202000$	
$L_1' = 00202000$	$R_1' = 00000600$	$p = \frac{10.12}{64.64}$
$L_2' = 00000600$	$R_2' = 00000000$	$p = \frac{10}{64}$
$L_3' = 00000000$	$R_3' = 00000600$	$p = 1$
$L_4' = 00000600$	$R_4' = 00202000$	$p = \frac{10}{64}$
$L_5' = 00202000$	$R_5' = 04040780$	$p = \frac{10.12}{64.64}$

9 Conclusions

The security of iterated ciphers and hash functions has been active and focussed research area for several years. DES is one of the widely known symmetric cryptosystem. In differential cryptanalysis, which has attracted a lot of researchers throughout the world in the area of cryptography, the main task is to study the propagation of differences from round to round inside the cipher and find specific differences, which propagate with relatively high probability. Such pairs of input-output differences can be used to recover some bits of the secret key. Here, in this paper, we have applied this cryptanalysis technique to DES reduced to 3-round, 4-round, 5-round and 6-round where we have differentiated between wrong and right pairs so that we can discard wrong pairs to get relevant key bits and finally could able to retrieve the correct key. Further, we performed the attack procedure for reduced 7-round and 8-round. Our cryptanalysis presented in this paper will surely be beneficial for extending the cryptanalysis procedure on further rounds.

References

Biham, E. and Shamir, A. (1990) *Differential Cryptanalysis of DES-like Cryptosystems*, July, The Weizmann Institute of Science Department of Apllied Mathematics.

Biham, E. and Shamir, A. (1993) *Differential Cryptanalysis of Data Encryption Standards*, 1st ed., Springer-Verlag New York, Inc.

Chun, K., Kim, S., Lee, S., Sung, S.H. and Yoon, S. (2002) 'Differential and linear cr yptanalysis for 2-round SPNs', *Information Processing Letters*, September , Vol. 87, No. 5, pp.277–282, Elsevier.

Davies, D.W. (1987) *Private Communication* [online] https://books.google.co.in/books?id=hy7jBwAAQBAJ&pg=PA183&lpg=PA183&dq=D.+W.+Davies,+Private+communications.

Den Boer, B. (1988) 'Cryptanalysis of F.E.A.L', *Advances in Cryptology, Proceedings of EUROCRYPT 88*, pp.293–300.

Feistel, H. (1973) 'Cryptography and computer privacy', *Scientific American*, Vol. 228, No. 5, pp.15–23.

Hellman, M.E., Merkle, R., Schroppei, R., Washington, L., Diffie, W., Pohlig, S. and Schweitzer, P. (1976) *Results of an Initial Attempt to Cryptanalyze the NBS Data Encryption Standard*, September, Standford University.

Heys, H.M. (2002) 'A tutorial on linear and differential cryptanalysis', *Journal Cryptologia*, Vol. 26, No. 3 [online] http://citeseer.nj.nec.com/443539.html.

Heys, H.M. and Tavares, S.E. (1996) 'Substitution-permutation networks resistant to differential and linear cryptanalysis', *Journal of Cryptology*, Vol. 9, No. 1, pp.1–19.

Miyaguchi, S., Shiraishi, A. and Shimizu, A. (1998) 'Fast data encryption algorithm Feal-8', *Review of Electrical Communications Laboratories*, Vol. 36, No. 4, pp.433–437.

National Bureau of Standards (1977) *Data Encryption Standard*, FIPS Publication No. 46, January, US Department of Commerce.

Schaumuller-Bichl, I. (1981) *Zur Analyse des Data Encryption Standard und Synthese Verwandter Chiffriersysteme*, PhD thesis, May, Linz University.

Schaumuller-Bichl, I. (1982) 'Cryptanalysis of the data encryption standard by the method of formal coding', *Cryptologia, Proceedings of CRYPTO 82*, pp.235–255.

Schaumuller-Bichl, I. (1983) *On the Design and Analysis of New Cipher Systems Related to the DES*, Technical report, Linz University.

Shimizu, A. and Miyaguchi, S. (1987a) 'Fast data encryption algorithm FEAL', *Abstracts of EUROCRYPT 87*, April, pp.VII-11–VII-14.

Shimizu, A. and Miyaguchi, S. (1987b) 'Fast data encryption algorithm FEAL', *Advances in Cryptology, Proceedings of EUROCRYPT 87*, pp.267–278.

Stallings, W. (2006) *Cryptography Theory and Network Security*, 4th ed., Pearson Education, USA.

Stinson, D.R. (1995) *Cryptography Theory and Practice*, 1st ed., CRC Press, UK.

Stinson, D.R. (2006) *Cryptography Theory and Practice*, 3rd ed., Chapman Hall/CRC, UK.

Tiwari, V., Venkaiah, V.C. and Tentu, A.N. (2017) 'Differential cryptanalysis of six rounds on data encryption standard: an implementation', *International Conference on Cyber Security (ICCS), Proceedings, IJASCSE*, August 12–13, RTU, Kota Rajasthan, Vol. 6, No. 9.