[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy
Scribed by: Archit Agrawal (202051213)

Winter 2022-2023
Lecture 5 and 6 (Week 3)

1 Quick Recap

In the second week, the following topics were covered:

- Classical Ciphers: Playfair, Hill and Substitution Cipher
- Kerkchoff's Principle and Shannon's Notion of Perfect Secrecy
- Symmetric Key Cipher: Block and Stream Cipher
- Product Cipher: Substitution Permutation Network
- Feistel Network
- Iterated Block Cipher
- One Time Padding

2 One Time Padding

One Time Padding (OTP) provides perfect secrecy under some conditions.

Encryption:

$$\begin{gathered} \text{P} \rightarrow \text{Plain Text} \\ \text{K} \rightarrow \text{Secret Key} \\ Enc(P, K) = P \oplus K = C \\ \oplus \rightarrow \text{xor operation} \end{gathered}$$

Decryption:

$$Dec(C, K) = C \oplus K = P$$

The conditions under which OTP provides perfect secrecy are as follows:

1. The secret key K cannot be used to encrypt two messages, that is, the key can not be reused.
2. The length of key must be greater than or equal to the length of message.
3. Key K is uniformly selected from the key space.

Stream Cipher is a generalisation of OTP which does not provide perfect secrecy.

2.1 OTP on One Bit Encryption

The message and key will be of one bit. The probabilities of the message and keys are mentioned below:

$$Pr[m = 0] = p$$
$$Pr[m = 1] = 1 - p$$
$$Pr[k = 0] = 0.5$$
$$Pr[k = 1] = 0.5$$

where m is one bit message ($m \in \{0, 1\}$) and k ($k \in \{0, 1\}$) is one bit key. Since, key is to be chosen uniformly, hence, its probability is 0.5 for values 0 and 1.
Encryption: C = m $\oplus$ k

Cipher text can be 0 or 1. Let us find the probability of cipher text to be zero.

$$\Pr[\text{C} = 0] = \Pr[\text{m} = 0, \text{k} = 0] + \Pr[\text{m} = 1, \text{k} = 1]$$

The events [m = 0, k = 0] and [m = 1, k = 1] are mutually exclusive gives the cipher text 0. Hence, probability of cipher text to be zero is summation of both these probabilities. Also, the message bits and the key bits are independent of each other, that is, the value of one doesn't affect the other in any way. Hence, the above equation can be written as:

$$Pr[C = 0] = Pr[m = 0] \cdot Pr[k = 0] + Pr[m = 1] \cdot Pr[k = 1]$$
$$Pr[C = 0] = p \cdot 0.5 + (1 - p) \cdot 0.5$$
$$Pr[C = 0] = 0.5$$

Similarly, it can be found that Pr[C = 1] = 0.5. To prove that OTP provides perfect secrecy, we need to prove the following statement:

$$Pr[M = m|C = c] = Pr[M = m]$$

We know that:

$$Pr[A|B] = \frac{Pr[AB]}{Pr[B]}$$
$$Pr[AB] = Pr[B|A] \cdot Pr[A]$$

Therefore,

$$Pr[m = 0|c = 0] = \frac{Pr[m=0,c=0]}{Pr[c=0]}$$

$$Pr[m = 0|c = 0] = \frac{Pr[c=0|m=0] \cdot Pr[m=0]}{Pr[c=0]}$$

If the cipher text is 0, given that the message is 0, then key can have only one possible value (e.g. 0), which means:

$$Pr[c = 0|m = 0] = Pr[k = 0] = 0.5$$
$$\therefore Pr[m = 0|c = 0] = \frac{0.5 \cdot Pr[m=0]}{0.5}$$

$$\Pr[\text{m} = 0 \text{ — } \text{c} = 0] = \Pr[\text{m} = 0]$$

Hence, the equation for perfect secrecy is proved under the conditions that were mentioned earlier. The proof can be generalized for any number of bits.

In the proof above, we took the length of key equal to the message and the key to be uniformly selected from key space by assigning equal probability of 0.5. Now, lets try to understand what will happen if same key is used to encrypt two messages.

$$M_1 \oplus k = C_1$$
$$M_2 \oplus k = C_2$$

M_1 and M_2 are two messages encrypted using the same key and we have only the cipher text C_1 and C_2.

$$C_1 \oplus C_2 = (M_1 \oplus k) \oplus (M_2 \oplus k)$$
$$C_1 \oplus C_2 = M_1 \oplus M_2$$

Xoring the cipher texts gives us some additional knowledge about the messages. Let's say $C_1 \oplus C_2 =$ 00..101. This states that the MSB's of the two messages M_1 and M_2 are same while their LSB's are different. That is, the difference in cipher text will reveal the difference in messages.

Now, lets understand what will happen when length of key is strictly lesser than length of message. Let's say there is a 32 bit message and a 16 bit key. Now, to make XOR operation possible, the key must become at least equal to the length of message. There are several ways this can be done, two of them are mentioned below:

1. Add 16 0's or 1's in the key, maybe as LSB or MSB.

   key: 0000 0000 0000 0000 k

   where k is actual 16 bit key. The problem here is that 16 bits of the message will not be encrypted only. That is, we will get 16 bits of message without any encryption in the cipher text.

2. The other way is to repeat a certain number of key bits and add them either as LSB or MSB.

   $$P = P_1P_2.....P_n$$
   $$K = K_1K_2....K_lK_1K_2....K_t$$

   where, $n = l + p$, n is length of message, t is length of key ($t < n$) and first l bits of key are used again.

   $$C = (P_1 \oplus K_1)(P_2 \oplus K_2)...(P_l \oplus K_l)(P_{l+1} \oplus K_1)...(P_n \oplus K_t)$$
   $$C = C_1C_2....C_lC_{l+1}....C_n$$

   Now, if we perform $C_1 \oplus C_{l+1}$, we will get $(P_1 \oplus P_{l+1})$. Hence, cipher text will reveal the difference between first and $(l+1)^{th}$ bit of the message.

One Time Padding with above conditions satisfying, i.e. providing perfect secrecy, is not usable in practical life. This is because for communication the key will be different every time and also length of key will be as long as (if not more) the message. This means that there will be a mechanism to share the secret key securely. Hence, the same mechanism can be used to communicate the message, rather than the security key.

3 Data Encryption Standard (DES)

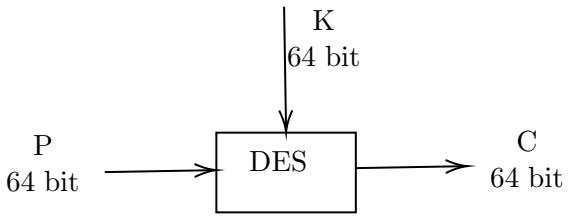
Data Encryption Standard is a block cipher designed by IBM. It is based on Feistel Networks. It has the following parameters:

- Block Size = 64 bits
- Number of Rounds = 16
- Secret Key Size = 64 bits (8 parity check bits)

Initially, the design was kept secret and it was used for personal communication only. But when the design came into the public domain, immediately the cipher was broken. There are several weaknesses which reveals the secret key in a quick time.

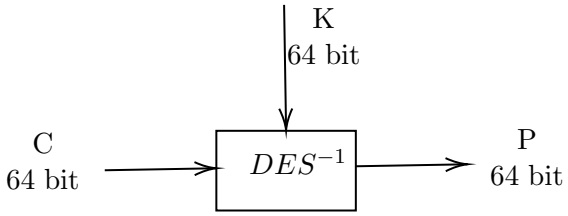
The actual secret key is if 56 bits. The 8 parity bits depends on the 56 bits of the key. The parity bits ensure if the other bits are correct or not.To find the secret key of DES exhaustively, 2^{56} keys will be checked in the worst case.

Encryption:



It takes 64 bit message and a 64 bit key and generates 64 bit cipher text.

Decryption:



The secret key is 64 bits long with every 8th bit (from MSB side) as a parity bit. For Example,

Key: 01101010 11010001 11010111

The red coloured bits are parity bits. These are calculated by xoring the 7 bits prior to each parity bit. Now, if there is an odd number of bits altered in the 7 bits, it can be identified using the parity bit. However, parity bits will not help in identifying an alteration in even number of bits.

The first step of DES is to discard these 8 parity bits and get the final secret key of 56 bits. Hence, DES should provide 56 bit security.

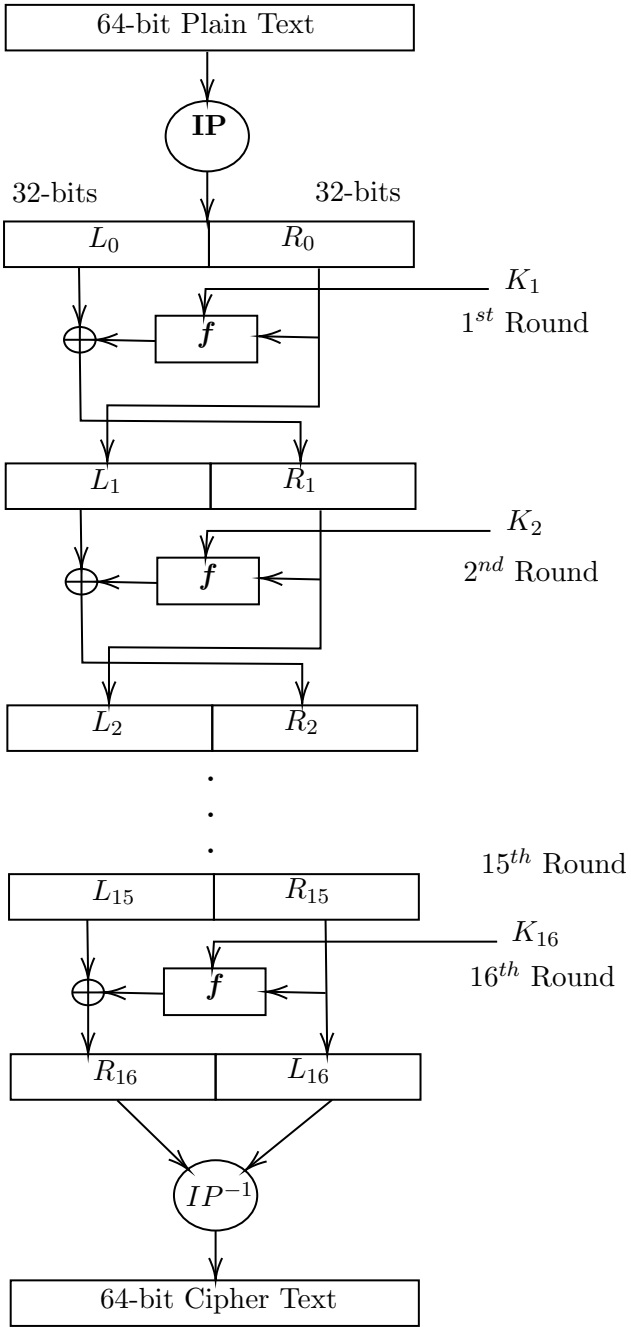
In DES, there are 16 rounds, hence, there will be a round function for every round. DES uses

the same round function in every round. For each round, there is a round key K_i which is generated by the key scheduling algorithm. It will take secret key as input and generate the round keys $K_1, K_2, ..., K_{16}$. The length of the round keys generated by the key scheduling algorithm is 48 bits.

The round function f for DES is defined as follows:

$$f : \{0,1\}^{32} \times \{0,1\}^{48} \rightarrow \{0,1\}^{32}$$

A flowchart for encrpyting a block of message is given below.



In the flowchart above, IP is the initial permutation and it will permute the message bits. Similar to Feistel Network, for each round L_{i+1} and R_{i+1} will be:

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus f(R_i, K_{i+1})$$

where $i \in \{0, 1, ..., 15\}$. In the last round, i.e. 16^{th} round, the position of L_{16} and R_{16} are swapped. After this round, the inverse of Initial Permutation (IP) is applied to the 64-bits and we get the cipher text.

We need to address the following now:

- Initial Permutation and its inverse
- The round function f
- How are the round keys $K_1, K_2, ..., K_{16}$ generated.

3.1 Initial Permutation of DES

It is a bijection from 64-bit to 64-bit. The 64-bits of the message are permuted and then the rounds are applied to the permuted message. Initial Permutation is defined as:

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

It can be interpreted as follows:

$$IP(m_1 m_2 ... m_7 m_8 m_9 .... m_{64}) = m_{58} m_{50} ... m_{10} m_2 m_{60} .... m_7$$

Using IP we can easily compute its inverse. The inverse of IP is provided in the Appendix

3.2 Round Function of DES

$$f : \{0,1\}^{32} \times \{0,1\}^{48} \to \{0,1\}^{32}$$
$$f(R_i, K_i) = X_i$$
where R_i is 32-bit, K_i is 48-bit and X_i is 32-bit.

The round function for DES is defined as:

$$f(R_i, K_i) = P(S(E(R_i) \oplus K_i))$$

Expansion Function E : $\{0,1\}^{32} \to \{0,1\}^{48}$
Substitution Box S: $\{0,1\}^{48} \to \{0,1\}^{32}$
Permutation Box P: $\{0,1\}^{32} \to \{0,1\}^{32}$

Hence,

length of R_i = 32-bits
length of $E(R_i)$ = 48-bits = length of K_i
length of $E(R_i) \oplus K_i$ = 48-bits
length of $S(E(R_i) \oplus K_i)$ = 32-bits
length of $P(S(E(R_i) \oplus K_i))$ = 32-bits

3.2.1 Expansion Function

$$E : \{0,1\}^{32} \rightarrow \{0,1\}^{48}$$

The expansion function for DES is given below:

$$E = \begin{bmatrix} 32 & 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 9 & 10 & 11 & 12 & 13 \\ 12 & 13 & 14 & 15 & 16 & 17 \\ 16 & 17 & 18 & 19 & 20 & 21 \\ 20 & 21 & 22 & 23 & 24 & 25 \\ 24 & 25 & 26 & 27 & 28 & 29 \\ 28 & 29 & 30 & 31 & 32 & 1 \end{bmatrix}$$

The bits are repeated for expanding 32-bits to 48-bits.

$$E(x_1 x_2 .... x_{32}) = (x_{32} x_1 x_2 x_3 x_4 x_5 x_4 x_5 .... x_{32} x_1)$$

3.2.2 Substitution Box

$$S : \{0,1\}^{48} \rightarrow \{0,1\}^{32}$$
$$S(X) = Y, \text{ where X is 48 and Y is 32 bit long}$$

Dividing X into 8 blocks each of length 6-bits.

$$X = B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8$$

Corresponding to each B_i there is a substitution box S_i where $i \in \{1, 2, .., 8\}$.

$$S_i : \{0,1\}^6 \rightarrow \{0,1\}^4 \forall i \in \{1, 2, .., 8\}$$
$$S_i(B_i) = C_i$$
$$\therefore S(X) = (S_1(B_1), S_2(B_2), S_3(B_3), S_4(B_4), S_5(B_5), S_6(B_6), S_7(B_7), S_8(B_8))$$

Therefore, length of S(X) is 32 bits. Now, let's see how to compute the output C_i given S_i and B_i. There is a 4×16 table corresponding to each S_i for DES. These tables are provided in the Appendix.

$$S_i = \begin{bmatrix} a_{0,0} & \dots & a_{0,15} \\ \vdots & \ddots & \vdots \\ a_{3,0} & \dots & a_{3,15} \end{bmatrix} \text{ where } a_{i,j} \in \{0, 1, ...15\}$$

Lets see how we have to look into these tables. Consider the binary representation of B_i.

$$B_i = b_1 b_2 b_3 b_4 b_5 b_6 \text{ where } b_i \in \{0, 1\}$$

Now, consider the MSB and LSB of B_i, i.e b_1 and b_6. The decimal value of the binary number $b_1 b_6$ will be:

$$r = 2 \cdot b_1 + b_6$$
$$\therefore r \in \{0, 1, 2, 3\}$$

r is called the row number. Now, consider the rest of the bits of B_i, i.e. $b_2 b_3 b_4 b_5$. The decimal value of $b_2 b_3 b_4 b_5$ (denoted by c) will be an integer in $0 \leq c \leq 15$. This c is called as the column number. The value of $S_i(B_i)$ is the entry in S_i table in row r and column c.

$$S_i(B_i) = a_{r,c}$$

3.2.3 Permutation Box

$$P : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$$

It is also defined by a table. The table is given below:

$$P = \begin{bmatrix} 16 & 7 & 20 & 21 \\ 29 & 12 & 28 & 17 \\ 1 & 15 & 23 & 26 \\ 5 & 18 & 31 & 10 \\ 2 & 8 & 24 & 14 \\ 32 & 27 & 3 & 9 \\ 19 & 13 & 30 & 6 \\ 22 & 11 & 4 & 25 \end{bmatrix}$$

$$P(x_1 x_2 ... x_{32}) = x_{16} x_7 x_{20} x_{21} x_{29} ... x_{25}$$

3.3 Key Scheduling Algorithm of DES

The Key Scheduling Algorithm generates round keys for each of the 16 rounds of DES. It takes the 64-bit secret key as input and produces 16 round keys, each of 48-bits.
The steps of the algorithm are given below:

- Define $v_i, 1 \leq i \leq 16$, where $v_i = 1$ if $i \in \{1, 2, 9, 16\}$, else $v_i = 2$.
- Discard 8 parity check bits from K. The 56 bit key is $\tilde{K}$.
- $T = PC1(\tilde{K})$, where PC1 is a permuation defined as:

  $$PC1 : \{0,1\}^{56} \rightarrow \{0,1\}^{56}$$

- $(C_0, D_0) = T$, where C_0 is most significant 28 bits of T and D_0 is least significant 28 bits of T.
- for $i = 1$ to 16:

  $$C_i = (C_{i-1} \hookleftarrow v_i)$$
  $$D_i = (D_{i-1} \hookleftarrow v_i)$$

  $\hookleftarrow$ is left circular shift. For Example: $x_1 x_2 ... x_{25} \hookleftarrow 2 = x_3 x_4 .... x_{25} x_1 x_2$.

  $$K_i = PC2(C_i, D_i)$$

  where, PC2 is a substitution defined as:

  $$PC2 : \{0,1\}^{56} \rightarrow \{0,1\}^{48}$$

- Round Keys = $\{K_1, K_2, ..., K_{16}\}$

3.3.1 PC1

$$PC1 : \{0,1\}^{56} \rightarrow \{0,1\}^{56}$$

$$\text{PC1} = \begin{bmatrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 \\ 1 & 58 & 50 & 42 & 34 & 26 & 18 \\ 10 & 2 & 59 & 51 & 43 & 35 & 27 \\ 19 & 11 & 3 & 60 & 52 & 44 & 36 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 \\ 7 & 62 & 54 & 46 & 38 & 30 & 22 \\ 14 & 6 & 61 & 53 & 45 & 37 & 29 \\ 21 & 13 & 5 & 28 & 20 & 12 & 4 \end{bmatrix}$$

Input to PC1 is the 56-bit key $\tilde{K}$ and the output is T.

$$T = PC1(K_1K_2...K_7K_9K_{10}....K_{63}) = K_{57}K_{49}....K_{12}K_4$$

The 28 MSB of T is C_i while 28 LSB is D_i.

3.3.2 PC2

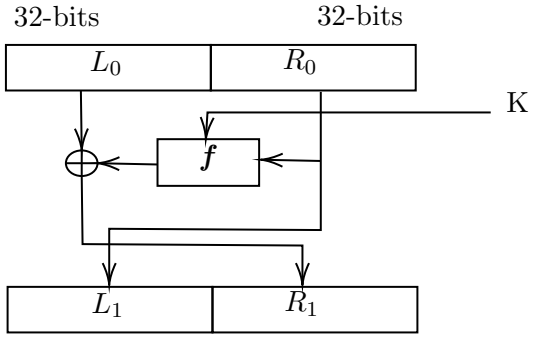
$$PC2 : \{0,1\}^{56} \rightarrow \{0,1\}^{48}$$

$$\text{PC2} = \begin{bmatrix} 14 & 17 & 11 & 24 & 1 & 5 \\ 3 & 28 & 15 & 6 & 21 & 10 \\ 23 & 19 & 12 & 4 & 26 & 8 \\ 16 & 7 & 27 & 20 & 13 & 2 \\ 41 & 52 & 31 & 37 & 47 & 55 \\ 30 & 40 & 51 & 45 & 33 & 48 \\ 44 & 49 & 39 & 56 & 34 & 53 \\ 46 & 42 & 50 & 36 & 29 & 32 \end{bmatrix}$$

$$K_i = PC2(C_i, D_i) = PC2(x_1x_2...x_{56}) = x_{14}x_{17}...x_{32}$$

4 Complementation Property of DES

Let us consider the Feistel Network structure agian.



Let us call this structure FN and it takes the message and key as input and generates the cipher text. The following are now known to us:

$$M = L_0||R_0 \text{ (the message)}$$
$$f = P(S(E(R_0) \oplus K))$$
$$C = L_1||R_1 \text{ (the cipher text)}$$

Now, consider the following two equations:

$$FN(M, K) = C_1$$
$$FN(\overline{M}, \overline{K}) = C_2$$

where $\overline{X}$ denotes bitwise complement of X. That is, if:

$$X = x_1x_2.....x_n$$
$$\overline{X} = (1 \oplus x_1)(1 \oplus x_2)....(1 \oplus x_n)$$

Let us try to find a relation between C_1 and C_2. Clearly,

$$L_{1M} = R_{0M} \text{ and } L_{1\overline{M}} = R_{0\overline{M}}$$

R_{0M} and $R_{0\overline{M}}$ are bit-wise complementary to each other ($\because$ M and $\overline{M}$ are complementary). Therefore, L_{1M} and $L_{1\overline{M}}$ are complementary.

Now, let's try to find the relation between $f(M, K)$ and $f(\overline{M}, \overline{K})$. Let's say $R_0 = x_1x_2....x_{32}$. Therefore,

$$\overline{R_0} = \overline{x_1x_2}...\overline{x_{32}}$$
$$\implies E(\overline{R_0}) = \overline{x_{32}x_1}...\overline{x_1}$$
$$\implies E(\overline{R_0}) = \overline{x_{32}x_1...x_1}$$
$$\implies E(\overline{R_0}) = \overline{E(R_0)}$$

Now, from the definition of round function f:

$$f(\overline{M}, \overline{K}) = P(S(E(R_{0\overline{M}}) \oplus \overline{K}))$$
$$f(\overline{M}, \overline{K}) = P(S(E(\overline{R_{0M}}) \oplus \overline{K}))$$
(because M and $\overline{M}$ are complementary)
$$f(\overline{M}, \overline{K}) = P(S(\overline{E(R_{0M})} \oplus \overline{K})) \text{ (proved above)}$$
$$f(\overline{M}, \overline{K}) = P(S(E(R_{0M}) \oplus K))$$
($\because A \oplus B = \overline{A} \oplus \overline{B}$)
$$f(\overline{M}, \overline{K}) = f(M, K)$$

Since, S and P are substituion and permuation respectively, they will generate same output if given the same input. Now,

$$R_{1M} = L_{0M} \oplus f(M, K) \text{ and } R_{1\overline{M}} = L_{0\overline{M}} \oplus f(\overline{M}, \overline{K})$$
$$\implies R_{1M} = L_{0M} \oplus f(M, K) \text{ and } R_{1\overline{M}} = L_{0\overline{M}} \oplus f(M, K)$$

Since, L_{0M} and $L_{0\overline{M}}$ are complementary, R_{1M} and $R_{1\overline{M}}$ should also be complementary. Since,

$$C_2 = L_{1\overline{M}}||R_{1\overline{M}}$$
$$C_2 = \overline{L_{1M}}||\overline{R_{1M}}$$
$$C_2 = \overline{L_{1M}||R_{1M}}$$
$$C_2 = \overline{C_1}$$

Hence, for Feistel Networks, the outputs will be complement to each other, if the messages and keys are complementary to each other.

The key scheduling algorithm for DES performs a permutation, a left circular shift and then a substitution. Since, these operations are exactly similar for each input, therefore, if two complementary inputs are given to the key scheduling algorithm:

- Left Circular Shift will generate complemented outputs.
- PC1 will generate complemented outputs as it is a fixed permutation of input.
- PC2 will generate complemented outputs as it is a fixed substituion of input.

Hence, the round keys generated for K and $\overline{K}$ will also be complementary.

If we encrypt two messages using DES, one M using key K and other $\overline{M}$ using key $\overline{K}$.

- The result after Initial Permutation will be complement to each other as it is a fixed permuation. This output is input to first round of feistel network.
- The input to first round of Feistel Network is complementary to each other. Also, the key scheduling algorithm will generate complementary keys. Hence, the outputs after first round of Feistel Network will be complementary. This will be input to second round of feistel network.
- Second round keys will be complementary, hence, second round will also generate complementary output and this will continue till the last round.
- The output is complementary after the 16 rounds and the final cipher text is generated by IP^{-1}, which again is a fixed permutation. Hence, the cipher text will also be complementary.

5 Appendix

The inverse of the Initial Permuation required in last step of DES is given as:

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

The substitution boxes S_i corresponding to each B_i for round function of DES are given below:

$$S_1 = \begin{bmatrix} 14 & 4 & 13 & 1 & 2 & 15 & 11 & 8 & 3 & 10 & 6 & 12 & 5 & 9 & 0 & 7 \\ 0 & 15 & 7 & 4 & 14 & 2 & 13 & 1 & 10 & 6 & 12 & 11 & 9 & 5 & 3 & 8 \\ 4 & 1 & 14 & 8 & 13 & 6 & 2 & 11 & 15 & 12 & 9 & 7 & 3 & 10 & 5 & 0 \\ 15 & 12 & 8 & 2 & 4 & 9 & 1 & 7 & 5 & 11 & 3 & 14 & 10 & 0 & 6 & 13 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 15 & 1 & 8 & 14 & 6 & 11 & 3 & 4 & 9 & 7 & 2 & 13 & 12 & 0 & 5 & 10 \\ 3 & 13 & 4 & 7 & 15 & 2 & 8 & 14 & 12 & 0 & 1 & 10 & 6 & 9 & 11 & 5 \\ 0 & 14 & 7 & 11 & 10 & 4 & 13 & 1 & 5 & 8 & 12 & 6 & 9 & 3 & 2 & 15 \\ 13 & 8 & 10 & 1 & 3 & 15 & 4 & 2 & 11 & 6 & 7 & 12 & 0 & 5 & 14 & 9 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 10 & 0 & 9 & 14 & 6 & 3 & 15 & 5 & 1 & 13 & 12 & 7 & 11 & 4 & 2 & 8 \\ 13 & 7 & 0 & 9 & 3 & 4 & 6 & 10 & 2 & 8 & 5 & 14 & 12 & 11 & 15 & 1 \\ 13 & 6 & 4 & 9 & 8 & 15 & 3 & 0 & 11 & 1 & 2 & 12 & 5 & 10 & 14 & 7 \\ 1 & 10 & 13 & 0 & 6 & 9 & 8 & 7 & 4 & 15 & 14 & 3 & 11 & 5 & 2 & 12 \end{bmatrix}$$

$$S_4 = \begin{bmatrix} 7 & 13 & 14 & 3 & 0 & 6 & 9 & 10 & 1 & 2 & 8 & 5 & 11 & 12 & 4 & 15 \\ 13 & 8 & 11 & 5 & 6 & 15 & 0 & 3 & 4 & 7 & 2 & 12 & 1 & 10 & 14 & 9 \\ 10 & 6 & 9 & 0 & 12 & 11 & 7 & 13 & 15 & 1 & 3 & 14 & 5 & 2 & 8 & 4 \\ 3 & 15 & 0 & 6 & 10 & 1 & 13 & 8 & 9 & 4 & 5 & 11 & 12 & 7 & 2 & 14 \end{bmatrix}$$

$$S_5 = \begin{bmatrix} 2 & 12 & 4 & 1 & 7 & 10 & 11 & 6 & 8 & 5 & 3 & 15 & 13 & 0 & 14 & 9 \\ 14 & 11 & 2 & 12 & 4 & 7 & 13 & 1 & 5 & 0 & 15 & 10 & 3 & 9 & 8 & 6 \\ 4 & 2 & 1 & 11 & 10 & 13 & 7 & 8 & 15 & 9 & 12 & 5 & 6 & 3 & 0 & 14 \\ 11 & 8 & 12 & 7 & 1 & 14 & 2 & 13 & 6 & 15 & 0 & 9 & 10 & 4 & 5 & 3 \end{bmatrix}$$

$$S_6 = \begin{bmatrix} 12 & 1 & 10 & 15 & 9 & 2 & 6 & 8 & 0 & 13 & 3 & 4 & 14 & 7 & 5 & 11 \\ 10 & 15 & 4 & 2 & 7 & 12 & 9 & 5 & 6 & 1 & 13 & 14 & 0 & 11 & 3 & 8 \\ 9 & 14 & 15 & 5 & 2 & 8 & 12 & 3 & 7 & 0 & 4 & 10 & 1 & 13 & 11 & 6 \\ 4 & 3 & 2 & 12 & 9 & 5 & 15 & 10 & 11 & 14 & 1 & 7 & 6 & 0 & 8 & 13 \end{bmatrix}$$

$$S_7 = \begin{bmatrix} 4 & 11 & 2 & 14 & 15 & 0 & 8 & 13 & 3 & 12 & 9 & 7 & 5 & 10 & 6 & 1 \\ 13 & 0 & 11 & 7 & 4 & 9 & 1 & 10 & 14 & 3 & 5 & 12 & 2 & 15 & 8 & 6 \\ 1 & 4 & 11 & 13 & 12 & 3 & 7 & 14 & 10 & 15 & 6 & 8 & 0 & 5 & 9 & 2 \\ 6 & 11 & 13 & 8 & 1 & 4 & 10 & 7 & 9 & 5 & 0 & 15 & 14 & 2 & 3 & 12 \end{bmatrix}$$

$$S_8 = \begin{bmatrix} 13 & 2 & 8 & 4 & 6 & 15 & 11 & 1 & 10 & 9 & 3 & 14 & 5 & 0 & 12 & 7 \\ 1 & 15 & 13 & 8 & 10 & 3 & 7 & 4 & 12 & 5 & 6 & 11 & 0 & 14 & 9 & 2 \\ 7 & 11 & 4 & 1 & 9 & 12 & 14 & 2 & 0 & 6 & 10 & 13 & 15 & 3 & 5 & 8 \\ 2 & 1 & 14 & 7 & 4 & 10 & 8 & 13 & 15 & 12 & 9 & 0 & 3 & 5 & 6 & 11 \end{bmatrix}$$