# بررسي الگوريتم DES

**مقدمه :**

الگوريتم DES = data encryption standard ، در دهه ي ۷۰ ميلادي در آمريكا بعنوان يك استاندارد كدگذاري مطرح شد و هنوز هم كاربرد دارد.

الگوريتم DES داده هايي به طول ۶۴ بيت را با استفاده از يك كليد ۶۴ بيتي كدگذاري مي كند (هر چند بدلايلي كه توضيح داده خواهد شد تنها از ۵۶ بيت اين كليد استفاده مؤثر مي شود ).
در اين الگوريتم يك بلاك ۶۴ بيتي از داده ها (plaintext) دريافت شده و سپس يك بلاك ۶۴ بيتي كدگذاري شده تحويل داده مي شود (cipher text) .

اين الگوريتم از ۱۶ مرحله (راند!) تشكيل شده است. يعني اينكه الگوريتم اصلي آن براي توليد داده ي كدگذاري شده ، ۱۶ بار تكرار مي شود.

بهترين راه براي درك الگوريتم آن حل يك مثال است:

فرض كنيد (اعداد داده شده در مبناي ۱۶ هستند) :

Key = 13 34 57 79 9B BC DF F1
Input data = 01 23 45 67 89 AB CD EF

فرض كنيد به داده ي ورودي يا همان پيغامي كه بايد رمزگذاري شود M بگوييم و به كليد ، K .

M مثال ما در حالت دو دويي به صورت زير است:

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

اكنون اين پيغام به دو نيمه ۳۲ بيتي چپ و راست تقسيم مي شود، يعني :

**L** = 0000 0001 0010 0011 0100 0101 0110 0111
**R** = 1000 1001 1010 1011 1100 1101 1110 1111

كليد مثال ما نيز در مبناي دو به صورت زير است :

**K** = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

و اما نكته اي در مورد كليد فوق. در الگوريتم DES ، هر هشتمين بيت مربوط به كليد ، بكار برده نمي شود. يعني بيت هاي شماره ۸ ، ۱۶ ، ۲۴ ، ۳۲ ، ۴۰ ، ۴۸ ، ۵۶ و ۶۴ درنظر گرفته نمي شوند (به همين جهت طول مؤثر كليد ۵۶ بيت است).

اكنون الگوريتم DES شروع مي شود:

**مرحله اول** : ايجاد ۱۶ زير كليد (subkey) هر كدام به طول ۴۸ بيت.

قبل از شروع اين قسمت، آشنايي با Permuted Choice 1 كه به صورت مخفف به آن PC-1 هم گفته مي شود ، لازم است:

**PC-1**

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

بوسيله ي اين جدول، جايگرداني بيت ها رخ مي دهد.
اولين عدد جدول فوق همانطور كه ملاحظه مي نماييد ۵۷ است. يعني اينكه ۵۷ امين بيت كليد اصلي به اولين بيت كليد جايگرداني شده (كه به آن K+مي گوييم) تبديل مي شود. سپس ۴۹ امين بيت كليد اصلي به دومين بيت كليد جايگرداني شده تبديل مي شود و همين طور الي آخر.
بنابراين با توجه به اعداد مثال فوق داريم :

**K** = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001
K--->(PC-1)--->K+
**K**+ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

پايان قسمت شيرين اول!

**بررسي الگوريتم DES- قسمت دوم**

در قسمت قبل به محاسبه ي زير رسيديم :

**K**+ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

سپس K+ نيز به دو نيمه ي ۲۸ بيتي زير تقسيم مي شود:

C0 = 1111000 0110011 0010101 0101111
D0 = 0101010 1011001 1001111 0001111

همانطور كه در قسمت قبل گفته شد ، در اين مرحله ۱۶ ساب كي (Cn,Dn) بايد ايجاد شود (در اينجا n از يك شروع مي شود تا ۱۶). اين ۱۶ زير كليد از C0 و D0 فوق به شرح زير توليد مي شوند:

ابتدا جدول زير را درنظر بگيريد:

| Iteration Number | Number of Left Shifts |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 2 |
| 9 | 1 |
| 10 | 2 |
| 11 | 2 |
| 12 | 2 |
| 13 | 2 |
| 14 | 2 |
| 15 | 2 |
| 16 | 1 |

**ياد آوري :**

left shift و يا شيفت چپ (در اينجا به صورت ويژه) به اين معنا است كه هر بيت يك مكان به سمت چپ انتقال و شيفت پيدا مي كند (منهاي اولين بيت كه به انتهاي بلاك منتقل خواهد شد) . براي مثال اگر 1111000011001100101010101111 يك بيت به سمت چپ شيفت داده شود معادل با 1110000110011001010101011111 خواهد بود.

معناي جدول فوق اين است كه براي مثال C1,D1 با يك شيفت به چپ C0,D0 توليد مي شوند. سپس C2,D2 با يك شيفت به چپ از C1,D1 ايجاد مي شوند (هر زير كليد جديد از زير كليد قبلي شيفت داده شده مطابق جدول فوق توليد مي شود) و الي آخر.

خوب! بر مبناي C0,D0 محاسبه شده و جدول شيفت هاي فوق داريم :

***C0*** = 1111000011001100101010101111
***D0*** = 0101010101100110011110001111

***C1*** = 1110000110011001010101011111
***D1*** = 1010101011001100111100011110

***C2*** = 1100001100110010101010111111
***D2*** = 0101010110011001111000111101

***C3*** = 0000110011001010101011111111
***D3*** = 0101011001100111100011110101

***C4*** = 0011001100101010101111111100
***D4*** = 0101100110011110001111010101

***C5*** = 1100110010101010111111110000
***D5*** = 0110011001111000111101010101

***C6*** = 0011001010101011111111000011
***D6*** = 1001100111100011110101010101

***C7*** = 1100101010101111111100001100
***D7*** = 0110011110001111010101010110

***C8*** = 0010101010111111110000110011
***D8*** = 1001111000111101010101011001

***C9*** = 0101010101111111100001100110
***D9*** = 0011110001111010101010110011
***C10*** = 0101010111111110000110011001
***D10*** = 1111000111101010101011001100

***C11*** = 0101011111111000011001100101
***D11*** = 1100011110101010101100110011

***C12*** = 0101111111100001100110010101
***D12*** = 0001111010101010110011001111

***C13*** = 0111111110000110011001010101
***D13*** = 0111101010101011001100111100

***C14*** = 1111111000011001100101010101
***D14*** = 1110101010101100110011110001

***C15*** = 1111100001100110010101010111
***D15*** = 1010101010110011001111000111

***C16*** = 1111000011001100101010101111
***D16*** = 0101010101100110011110001111

اكنون نوبت اعمال PC-2بر روي جفت هاي فوق است تا Kn ها تشكيل شوند. اين جدول جايگرداني شماره ۲ به صورت زير است :

**PC-2**

| | | | | | |
|---|---|---|---|---|---|
| 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 |

براي مثال از محاسبات فوق داريم :

***C1D1*** = 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110

اكنون PC-2 بايد بر روي تك تك ۱۶ زير كليد محاسبه شده به صورت مستقل اعمال شود. يعني براي مثال در K1 ، (با توجه به جدول فوق) اولين بيت، معادل ۱۴ امين بيت C1D1 است، دومين بيت K1 معادل ۱۷ امين بيت C1D1 است و الي آخر.

براي صرفه جويي در وقت، پس از اعمال PC-2 بر روي CnDn هاي محاسبه شده فوق داريم:

***K1*** = 000110 110000 001011 101111 111111 000111 000001 110010
***K2*** = 011110 011010 111011 011001 110110 111100 100111 100101
***K3*** = 010101 011111 110010 001010 010000 101100 111110 011001
***K4*** = 011100 101010 110111 010110 110110 110011 010100 011101
***K5*** = 011111 001110 110000 000111 111010 110101 001110 101000
***K6*** = 011000 111010 010100 111110 010100 000111 101100 101111
***K7*** = 111011 001000 010010 110111 111101 100001 100010 111100
***K8*** = 111101 111000 101000 111010 110000 010011 101111 111011
***K9*** = 111000 001101 101111 101011 111011 011110 011110 000001
***K10*** = 101100 011111 001101 000111 101110 100100 011001 001111
***K11*** = 001000 010101 111111 010011 110111 101101 001110 000110
***K12*** = 011101 010111 000111 110101 100101 000110 011111 101001
***K13*** = 100101 111100 010111 010001 111110 101011 101001 000001
***K14*** = 010111 110100 001110 110111 111100 101110 011100 111010
***K15*** = 101111 111001 000110 001101 001111 010011 111100 001010
***K16*** = 110010 110011 110110 001011 000011 100001 011111 110101

پايان قسمت شيرين دوم!

**بررسي الگوريتم DES- قسمت سوم**

اگر بخواهيم تا اينجا قسمت تهيه ساب كي ها را خلاصه كنيم به صورت زير مي شود :

```
C[0]D[0] = PC1(key)
for 1 <= i <= 16
C[i] = LS[i](C[i-1])
D[i] = LS[i](D[i-1])
K[i] = PC2(C[i]D[i])
```

از اين قسمت به بعد كدگذاري بلاك هاي داده ي ۶۴ بيتي ما آغاز مي شود (DES Core Function) .

جدول زير را درنظر بگيريد:

**IP**

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

جدول IPدر اينجا به معناي initial permutationاست (جايگرداني اوليه) و بر روي پيغام اوليه ما يعني M، اعمال مي شود.

**M** = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
**IP** = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

همانند روش هاي اعمال جداول قبل، ۵۸ امين بيت M كه در اينجا "1" است اولين بيت IP مي شود، سپس ۵۰ امين بيت M كه در اينجا "1" است ، دومين بيت IP مي گردد و الي آخر.

سپس IP به دو نيمه ي ۳۲ بيتي چپ و راست تقسيم مي شود :

***L0*** = 1100 1100 0000 0000 1100 1100 1111 1111
***R0*** = 1111 0000 1010 1010 1111 0000 1010 1010

فرمول كلي اين ۱۶ راند به صورت زير است :

```
L[n] = R[n-1]
R[n] = L[n-1] XOR f(R[n-1],K[n])
```

در اينجا nبين ۱ و ۱۶ (راند) تغيير مي كند . در مورد تابع fدر ادامه توضيح داده خواهد شد.
براي مثال براي n=1داريم :

مواردي كه تا اينجا محاسبه شدند:

L[0] = 1100 1100 0000 0000 1100 1100 1111 1111
R[0] = 1111 0000 1010 1010 1111 0000 1010 1010

K[1] = 000110 110000 001011 101111 111111 000111 000001 110010

سپس :

L[1] = R[0] = 1111 0000 1010 1010 1111 0000 1010 1010
R[1] = L[0] + f(R[0],K[1])

قسمت باقيمانده ، توضيح عملكرد تابع fدر معادله ي فوق است:

براي محاسبه ي f ابتدا بايد R[n-1 ] سي و دو بيتي به ۴۸ بيت بسط داده شود. براي انجام اينكار از يك جدول انتخاب (selection table) به نام E استفاده مي شود:

**E**(**R[n-1**])

تابع فوق ورودي اش ۳۲ بيتي بوده و خروجي اش ۴۸ بيتي است. اين جدول به صورت زير است:

**E BIT-SELECTION TABLE**

| | | | | | |
|---|---|---|---|---|---|
| 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 |

محاسبه ي تابع f ادامه دارد...

پايان قسمت (شيرين) سوم!

**بررسي الگوريتم DES- قسمت چهارم**

در ادامه قسمت قبل فرض كنيد مي خواهيم E(R[0]) را از R[0] محاسبه نماييم :

R[0] = 1111 0000 1010 1010 1111 0000 1010 1010
E(R[0]) = 011110 100001 010101 010101 011110 100001 010101 010101

همانطور كه ملاحظه مي كنيد با اعمال جدول E ، هر ۴ بيت اوليه به ۶ بيت بسط داده شده است.

در ادامه ي محاسبه f، خروجي E(R[n-1])با كليد K[n]، XORمي شود. براي مثال :

K[1] = 000110 110000 001011 101111 111111 000111 000001 110010
E(R[0]) = 011110 100001 010101 010101 011110 100001 010101 010101
K1 XOR E(R[0]) = 011000 010001 011110 111010 100001 100110 010100 100111

هنوز محاسبه ي f تمام نشده است. تا اينجا R[n-1] از ۳۲ بيت به ۴۸ بيت با استفاده از جدول E ، بسط يافته است. سپس حاصل آن با K[n] ، XOR شده است. اكنون ۴۸ بيت و يا هشت گروه ۶ بيتي حاصل كار است. به اين ۸ گروه نامهايي از B[1] تا B[8] نسبت مي دهيم. يعني :

K[n] XOR E(R[n]-1) =B[1] B[2] B[3] B[4] B[5] B[6] B[7] B[8]

در ادامه نوبت عملياتي ديگر بر روي اين B[i] ها است. در اينجا مفهومي ديگر به نام S boxes ارائه مي شود. اين جداول بايد بر روي B[i] ها اعمال گردند. يعني:

S1(B[1]) S2(B[2]) S3(B[3]) S4(B[4]) S5(B[5]) S6(B[6]) S7(B[7]) S8(B[8])

اين جداول به شرح زير هستند :

**S1**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

**S2**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

**S3**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

**S4**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

**S5**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

**S6**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

**S7**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

**S8**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

روش اعمال S boxes با جداول ديگر متفاوت است كه در قسمت بعد بررسي خواهد شد.

پايان قسمت شيرين چهارم!

**بررسي الگوريتم DES- قسمت پنجم**

نحوه ي استفاده از S-boxes و يا Substitution boxes :

فرض كنيد عدد ۴۸ بيتي باينري زير را داريم كه مي خواهيم S-Boxes را بر آن اعمال كنيم :

011101000101110101000111101000011100101101011101

همانطور كه در قسمت قبل گفته شد ، ۸ گروه ۶ بيتي از آن استخراج مي شود كه از B1 تا B8 را تشكيل مي دهند (از چپ به راست):

011101 000101 110101 000111 101000 011100 101101 011101

محاسبات زير را در نظر بگيريد:

B[n] => S[n][row][column]

B[1] => S[1](01, 1110) = S[1][1][14] = 3 = 0011
B[2] => S[2](01, 0010) = S[2][1][2 ] = 4 = 0100
B[3] => S[3](11, 1010) = S[3][3][10] = 14 = 1110
B[4] => S[4](01, 0011) = S[4][1][3 ] = 5 = 0101
B[5] => S[5](10, 0100) = S[5][2][4 ] = 10 = 1010
B[6] => S[6](00, 1110) = S[6][0][14] = 5 = 0101
B[7] => S[7](11, 0110) = S[7][3][6 ] = 10 = 1010
B[8] => S[8](01, 1110) = S[8][1][14] = 9 = 1001

نحوه ي محاسبه :

nدقيقا متناظر است با انديس B .
Row : از كنار هم قرار گرفتن بيت اول و بيت آخر يك گروه ۶ بيتي فوق تشكيل مي شود.
Column : مابقي بيت ها ، ستون را تشكيل مي دهند (يعني از بيت هاي ۲ تا ۵)

براي مثال :

011101را در نظر بگيريد.
چون اولين گروه ۶ بيتي عدد ۴۸ بيتي ما را تشكيل مي دهد ، n=1 خواهد بود.
براي تشكيل Row ، دو بيت اول و آخر كنار هم قرار مي گيرد، يعني Row=01 .
Column تشكيل شده از بيت ۲ تا ۵ عدد فوق است يعني : Column=1110

نتيجه ي حاصل :

اولين سطر محاسباتي است كه در بالا ذكر شد يعني :
S[n][row][column] = S[1](01, 1110)

خوب! براي محاسبه ي اين موقعيت در جدول S1 ، ابتدا اعداد درون پرانتزها به معادل دسيمال خود تبديل مي شوند يعني:

S[1][1][14]

جدول S1 هم به صورت زير است (براي سهولت مراجعه شماره رديف ها و ستون ها نيز نوشته شده است، صفر تا سه شماره رديف ها هستند و صفر تا ۱۵ شماره ستون ها):

**S1 (ROW/Column)**

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ۰ | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| ۱ | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | *3* | 8 |
| ۲ | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| ۳ | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

سپس اين موقعيت در جدول S1 پيدا مي شود. يعني به جدول S1 مراجعه كرده و سپس عدد قرار گرفته در سطر ۱ و ستون ۱۴ را پيدا مي كنيم. اين عدد معادل ۳ است. سپس آنرا به باينري تبديل مي نماييم : 0001 .
بنابراين به صورت خلاصه داريم :

B[1] => S[1](01, 1110) = S[1][1][14] = 3 = 0011

به همين ترتيب براي ساير گروه هاي ۶ بيتي عمل مي شود.

نتيجه اين قسمت (كنار هم قرار دادن نتايج حاصل فوق، كه يك عدد ۳۲ بيتي دودويي را تشكيل مي دهد):

00110100111001011010010110101001

مثالي ديگر:
در ادامه ي اعداد اصلي انتخاب شده براي اين توضيحات در قسمت هاي قبل :

K[1] XOR E(R[0]) = 011000 010001 011110 111010 100001 100110 010100 100111
==>
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]) = 0101 1100 1000 0010 1011 0101 1001 0111

پايان قسمت شيرين پنجم!

**بررسي الگوريتم DES- قسمت پنجم**

پس از اعمال جداول جانشيني و يا همان S-Boxes بر روي Bi ها، بر روي نتيجه ي حاصل يك جايگرداني ديگر صورت مي گيرد.

**Permutation P**

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

يعني به صورت خلاصه تابع f كه در چند قسمت قبل راجع به آن بحث شد به صورت زير محاسبه مي شود:

f = P(S[1](B[1])...S[8](B[8]))

براي مثال :

```
S1(B1)S2(B2)S3(B3)S4(B4)S5(B5)S6(B6)S7(B7)S8(B8) = 0101 1100 1000 0010 1011 0101 1001 0111
==>
f = 0010 0011 0100 1010 1010 1001 1011 1011
```

و در ادامه :

```
R[1] =  L[0]  XOR  f(R[0] , K[1] )
     =  1100  1100  0000  0000  1100   1100   1111   1111
   XOR  0010  0011  0100  1010  1010   1001   1011   1011
     =  1110  1111  0100  1010   0110   0101   0100   0100
```

و L1 هم كه قبلا محاسبه شده بود :

L[1] = R[0] = 1111 0000 1010 1010 1111 0000 1010 1010

خوب! تا اينجا يك راند از ۱۶ راند الگوريتم دس پايان پذيرفت!

شروع راند دوم!

مطابق فرمول كلي عنوان شده داريم :

```
L[2] = R[1]
R[2] = L[1] + f(R[1], K[2])
```

كه محاسبه ي آن با توجه به مطالب گفته شده تاكنون ساده است (بعنوان تمرين آنرا انجام دهيد!). اين رويه تا پايان ۱۶ راند ادامه دارد.

در پايان راند ۱۶ ، حاصل كار L16 و R16 است. در اينجا جاي اين دو بلاك معكوس مي شود يعني :

R[16]L[16]

و بر روي آن آخرين جايگرداني مطابق جدول زير صورت مي گيرد :

IP^-1

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

براي مثال با توجه به اعداد انتخاب شده در ابتداي اين مباحث داريم :

L16 = 0100 0011 0100 0010 0011 0010 0011 0100
R16 = 0000 1010 0100 1100 1101 1001 1001 0101
===>
R[16]L[16] = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100
===>
IP^-1 = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101 (bin)
= 85E813540F0AB405 (hex)

**و يا به صورت خلاصه :**

پيغام اوليه :

M = 0123456789ABCDEF

كليد بكار گرفته شده براي كدگذاري :

Key = 13 34 57 79 9B BC DF F1

خروجي كدگذاري شده :

C = 85E813540F0AB405

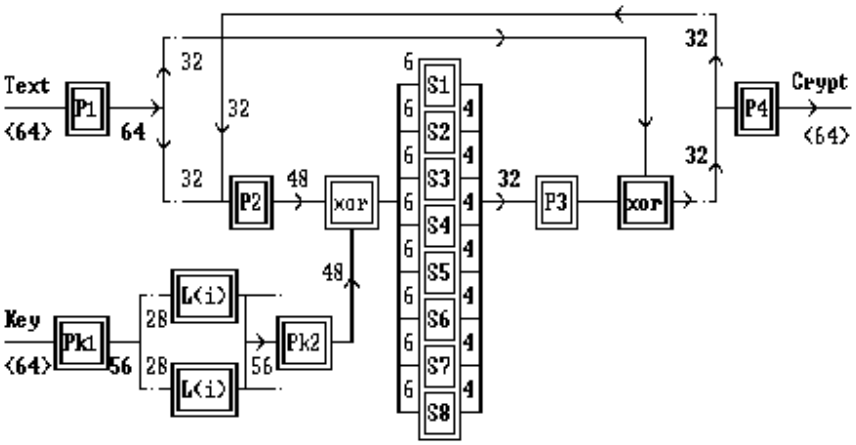
تا اينجا بررسي الگوريتم DESبه پايان مي رسد. در قسمت بعدي خلاصه ي محاسبات اين ۱۶ راند براي تكميل مبحث ارائه خواهد شد و همچنين تصاويري در مورد فلوچارت اين عمليات.

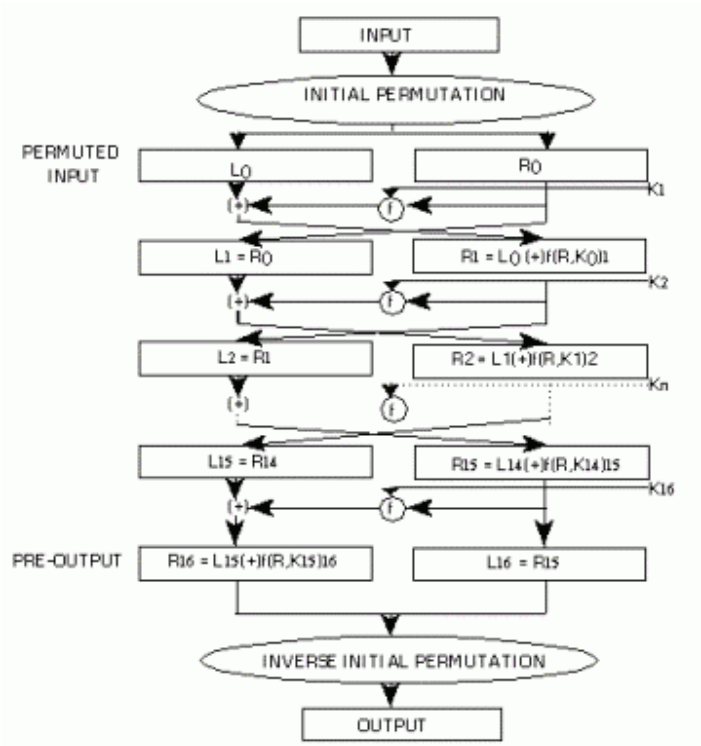
منابع و مآخذي كه براي تهيه اين مقالات مورد استفاده قرار گرفتند :

http://governmentsecurity.org/articles/AlgorithmsExplained.php
http://www.eventid.net/docs/desexample.htm
http://www.tropsoft.com/strongenc/des.htm
http://www.cryptography.com/resources/whitepapers/DES.html
http://www.cryptography.com/resources/whitepapers/DES-photos.html
http://www.cryptogram.org/cdb/aca.info/sample-issue.html
http://www.mapleapps.com/powertools/cryptography/HTML/DES-Example.html
http://www.orlingrabbe.com/des.htm
http://www.darkside.com.au/bitslice/index.html

**ضمايم:**

**الف) دو تصوير زير الگوريتم DES را به خوبي نمايش مي دهند :**



**ب) مثالي كامل از ۱۶ راند الگوريتم DES :**

**Key**= 13 34 57 79 9B BC DF F1
Key= 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001
**Input data** = 01 23 45 67 89 AB CD EF
Input data = 00000001 00100011 01000101 01100111 10001001 10101011 11001101 11101111
IP data= CC 00 CC FF F0 AA F0 AA
IP data= 11001100 00000000 11001100 11111111 11110000 10101010 11110000 10101010

---------- **Round 1** ----------
L= CC 00 CC FF
L= 11001100 00000000 11001100 11111111
R= F0 AA F0 AA
R= 11110000 10101010 11110000 10101010
K= 1B 02 EF FC 70
K= 00011011 00000010 11101111 11111100 01110000
----------F(R, K)----------
E(R)= 7A 15 55 7A 15
E(R)= 01111010 00010101 01010101 01111010 00010101
E(R) xor K = 61 17 BA 86 65
E(R) xor K = 01100001 00010111 10111010 10000110 01100101
SBoxed= 5C 82 B5 97
SBoxed= 01011100 10000010 10110101 10010111
P modification processed and result in F(R, K)...
---
F(R, K)= 23 4A A9 BB
F(R, K)= 00100011 01001010 10101001 10111011
R1= EF 4A 65 44
R1= 11101111 01001010 01100101 01000100
L1= F0 AA F0 AA
L1= 11110000 10101010 11110000 10101010

---------- **Round 2** ----------
L= F0 AA F0 AA
L= 11110000 10101010 11110000 10101010
R= EF 4A 65 44
R= 11101111 01001010 01100101 01000100
K= 79 AE D9 DB C9
K= 01111001 10101110 11011001 11011011 11001001
----------F(R, K)----------
E(R)= 75 EA 54 30 AA
E(R)= 01110101 11101010 01010100 00110000 10101010
E(R) xor K = 0C 44 8D EB 63
E(R) xor K = 00001100 01000100 10001101 11101011 01100011
SBoxed= F8 D0 3A AE
SBoxed= 11111000 11010000 00111010 10101110
P modification processed and result in F(R, K)...
---
F(R, K)= 3C AB 87 A3
F(R, K)= 00111100 10101011 10000111 10100011
R1= CC 01 77 09
R1= 11001100 00000001 01110111 00001001
L1= EF 4A 65 44
L1= 11101111 01001010 01100101 01000100

---------- **Round 3** ----------
L= EF 4A 65 44
L= 11101111 01001010 01100101 01000100

R= CC 01 77 09
R= 11001100 00000001 01110111 00001001
K= 55 FC 8A 42 CF
K= 01010101 11111100 10001010 01000010 11001111
----------F(R, K)----------
E(R)= E5 80 02 BA E8
E(R)= 11100101 10000000 00000010 10111010 11101000
E(R) xor K = B0 7C 88 F8 27
E(R) xor K = 10110000 01111100 10001000 11111000 00100111
SBoxed= 27 10 E1 6F
SBoxed= 00100111 00010000 11100001 01101111
P  modification processed and result in F(R, K)...
---
F(R, K)= 4D 16 6E B0
F(R, K)= 01001101 00010110 01101110 10110000
R1= A2 5C 0B F4
R1= 10100010 01011100 00001011 11110100
L1= CC 01 77 09
L1= 11001100 00000001 01110111 00001001

---------- **Round  4** ----------
L= CC 01 77 09
L= 11001100 00000001 01110111 00001001
R= A2 5C 0B F4
R= 10100010 01011100 00001011 11110100
K= 72 AD D6 DB 35
K= 01110010 10101101 11010110 11011011 00110101
----------F(R, K)----------
E(R)= 50 42 F8 05 7F
E(R)= 01010000 01000010 11111000 00000101 01111111
E(R) xor K = 22 EF 2E DE 4A
E(R) xor K = 00100010 11101111 00101110 11011110 01001010
SBoxed= 21 ED 9F 3A
SBoxed= 00100001 11101101 10011111 00111010
P  modification processed and result in F(R, K)...
---
F(R, K)= BB 23 77 4C
F(R, K)= 10111011 00100011 01110111 01001100
R1= 77 22 00 45
R1= 01110111 00100010 00000000 01000101
L1= A2 5C 0B F4
L1= 10100010 01011100 00001011 11110100

---------- **Round  5** ----------
L= A2 5C 0B F4
L= 10100010 01011100 00001011 11110100
R= 77 22 00 45
R= 01110111 00100010 00000000 01000101
K= 7C EC 07 EB 53
K= 01111100 11101100 00000111 11101011 01010011
----------F(R, K)----------
E(R)= BA E9 04 00 02
E(R)= 10111010 11101001 00000100 00000000 00000010
E(R) xor K = C6 05 03 EB 51
E(R) xor K = 11000110 00000101 00000011 11101011 01010001
SBoxed= 50 C8 31 EB
SBoxed= 01010000 11001000 00110001 11101011
P  modification processed and result in F(R, K)...
---

F(R, K)= 28 13 AD C3
F(R, K)= 00101000 00010011 10101101 11000011
R1= 8A 4F A6 37
R1= 10001010 01001111 10100110 00110111
L1= 77 22 00 45
L1= 01110111 00100010 00000000 01000101

---------- **Round 6** ----------
L= 77 22 00 45
L= 01110111 00100010 00000000 01000101
R= 8A 4F A6 37
R= 10001010 01001111 10100110 00110111
K= 63 A5 3E 50 7B
K= 01100011 10100101 00111110 01010000 01111011
----------F(R, K)----------
E(R)= C5 42 5F D0 C1
E(R)= 11000101 01000010 01011111 11010000 11000001
E(R) xor K = A6 E7 61 80 BA
E(R) xor K = 10100110 11100111 01100001 10000000 10111010
SBoxed= 41 F3 4C 3D
SBoxed= 01000001 11110011 01001100 00111101
P modification processed and result in F(R, K)...
---
F(R, K)= 9E 45 CD 2C
F(R, K)= 10011110 01000101 11001101 00101100
R1= E9 67 CD 69
R1= 11101001 01100111 11001101 01101001
L1= 8A 4F A6 37
L1= 10001010 01001111 10100110 00110111

---------- **Round 7** ----------
L= 8A 4F A6 37
L= 10001010 01001111 10100110 00110111
R= E9 67 CD 69
R= 11101001 01100111 11001101 01101001
K= EC 84 B7 F6 18
K= 11101100 10000100 10110111 11110110 00011000
----------F(R, K)----------
E(R)= F5 2B 0F E5 AB
E(R)= 11110101 00101011 00001111 11100101 10101011
E(R) xor K = 19 AF B8 13 B3
E(R) xor K = 00011001 10101111 10111000 00010011 10110011
SBoxed= 10 75 40 AD
SBoxed= 00010000 01110101 01000000 10101101
P modification processed and result in F(R, K)...
---
F(R, K)= 8C 05 1C 27
F(R, K)= 10001100 00000101 00011100 00100111
R1= 06 4A BA 10
R1= 00000110 01001010 10111010 00010000
L1= E9 67 CD 69
L1= 11101001 01100111 11001101 01101001

---------- **Round 8** ----------
L= E9 67 CD 69
L= 11101001 01100111 11001101 01101001
R= 06 4A BA 10
R= 00000110 01001010 10111010 00010000
K= F7 8A 3A C1 3B

K= 11110111 10001010 00111010 11000001 00111011
----------F(R, K)----------
E(R)= 00 C2 55 5F 40
E(R)= 00000000 11000010 01010101 01011111 01000000
E(R) xor K = F7 48 6F 9E 7B
E(R) xor K = 11110111 01001000 01101111 10011110 01111011
SBoxed= 6C 18 7C AE
SBoxed= 01101100 00011000 01111100 10101110
P modification processed and result in F(R, K)...
---
F(R, K)= 3C 0E 86 F9
F(R, K)= 00111100 00001110 10000110 11111001
R1= D5 69 4B 90
R1= 11010101 01101001 01001011 10010000
L1= 06 4A BA 10
L1= 00000110 01001010 10111010 00010000

---------- **Round 9** ----------
L= 06 4A BA 10
L= 00000110 01001010 10111010 00010000
R= D5 69 4B 90
R= 11010101 01101001 01001011 10010000
K= E0 DB EB ED E7
K= 11100000 11011011 11101011 11101101 11100111
----------F(R, K)----------
E(R)= 6A AB 52 A5 7C
E(R)= 01101010 10101011 01010010 10100101 01111100
E(R) xor K = 8A 70 B9 48 9B
E(R) xor K = 10001010 01110000 10111001 01001000 10011011
SBoxed= 11 0C 57 77
SBoxed= 00010001 00001100 01010111 01110111
P modification processed and result in F(R, K)...
---
F(R, K)= 22 36 7C 6A
F(R, K)= 00100010 00110110 01111100 01101010
R1= 24 7C C6 7A
R1= 00100100 01111100 11000110 01111010
L1= D5 69 4B 90
L1= 11010101 01101001 01001011 10010000

---------- **Round 10** ----------
L= D5 69 4B 90
L= 11010101 01101001 01001011 10010000
R= 24 7C C6 7A
R= 00100100 01111100 11000110 01111010
K= B1 F3 47 BA 46
K= 10110001 11110011 01000111 10111010 01000110
----------F(R, K)----------
E(R)= 10 83 F9 60 C3
E(R)= 00010000 10000011 11111001 01100000 11000011
E(R) xor K = A1 70 BE DA 85
E(R) xor K = 10100001 01110000 10111110 11011010 10000101
SBoxed= DA 04 52 75
SBoxed= 11011010 00000100 01010010 01110101
P modification processed and result in F(R, K)...
---
F(R, K)= 62 BC 9C 22
F(R, K)= 01100010 10111100 10011100 00100010
R1= B7 D5 D7 B2

R1= 10110111 11010101 11010111 10110010
L1= 24 7C C6 7A
L1= 00100100 01111100 11000110 01111010

---------- **Round 11** ----------
L= 24 7C C6 7A
L= 00100100 01111100 11000110 01111010
R= B7 D5 D7 B2
R= 10110111 11010101 11010111 10110010
K= 21 5F D3 DE D3
K= 00100001 01011111 11010011 11011110 11010011
----------F(R, K)----------
E(R)= 5A FE AB EA FD
E(R)= 01011010 11111110 10101011 11101010 11111101
E(R) xor K = 7B A1 78 34 2E
E(R) xor K = 01111011 10100001 01111000 00110100 00101110
SBoxed= 73 05 D1 01
SBoxed= 01110011 00000101 11010001 00000001
P modification processed and result in F(R, K)...
---
F(R, K)= E1 04 FA 02
F(R, K)= 11100001 00000100 11111010 00000010
R1= C5 78 3C 78
R1= 11000101 01111000 00111100 01111000
L1= B7 D5 D7 B2
L1= 10110111 11010101 11010111 10110010

---------- **Round 12** ----------
L= B7 D5 D7 B2
L= 10110111 11010101 11010111 10110010
R= C5 78 3C 78
R= 11000101 01111000 00111100 01111000
K= 75 71 F5 94 67
K= 01110101 01110001 11110101 10010100 01100111
----------F(R, K)----------
E(R)= 60 AB F0 1F 83
E(R)= 01100000 10101011 11110000 00011111 10000011
E(R) xor K = 15 DA 05 8B E4
E(R) xor K = 00010101 11011010 00000101 10001011 11100100
SBoxed= 7B 8B 26 35
SBoxed= 01111011 10001011 00100110 00110101
P modification processed and result in F(R, K)...
---
F(R, K)= C2 68 CF EA
F(R, K)= 11000010 01101000 11001111 11101010
R1= 75 BD 18 58
R1= 01110101 10111101 00011000 01011000
L1= C5 78 3C 78
L1= 11000101 01111000 00111100 01111000

---------- **Round 13** ----------
L= C5 78 3C 78
L= 11000101 01111000 00111100 01111000
R= 75 BD 18 58
R= 01110101 10111101 00011000 01011000
K= 97 C5 D1 FA BA
K= 10010111 11000101 11010001 11111010 10111010
----------F(R, K)----------
E(R)= 3A BD FA 8F 02

E(R)= 00111010 10111101 11111010 10001111 00000010
E(R) xor K = AD 78 2B 75 B8
E(R) xor K = 10101101 01111000 00101011 01110101 10111000
SBoxed= 9A D1 8B 4F
SBoxed= 10011010 11010001 10001011 01001111
P modification processed and result in F(R, K)...
---
F(R, K)= DD BB 29 22
F(R, K)= 11011101 10111011 00101001 00100010
R1= 18 C3 15 5A
R1= 00011000 11000011 00010101 01011010
L1= 75 BD 18 58
L1= 01110101 10111101 00011000 01011000

---------- **Round 14** ----------
L= 75 BD 18 58
L= 01110101 10111101 00011000 01011000
R= 18 C3 15 5A
R= 00011000 11000011 00010101 01011010
K= 5F 43 B7 F2 E7
K= 01011111 01000011 10110111 11110010 11100111
----------F(R, K)----------
E(R)= 0F 16 06 8A AA
E(R)= 00001111 00010110 00000110 10001010 10101010
E(R) xor K = 50 55 B1 78 4D
E(R) xor K = 01010000 01010101 10110001 01111000 01001101
SBoxed= 64 79 9A F1
SBoxed= 01100100 01111001 10011010 11110001
P modification processed and result in F(R, K)...
---
F(R, K)= B7 31 8E 55
F(R, K)= 10110111 00110001 10001110 01010101
R1= C2 8C 96 0D
R1= 11000010 10001100 10010110 00001101
L1= 18 C3 15 5A
L1= 00011000 11000011 00010101 01011010

---------- **Round 15** ----------
L= 18 C3 15 5A
L= 00011000 11000011 00010101 01011010
R= C2 8C 96 0D
R= 11000010 10001100 10010110 00001101
K= BF 91 8D 3D 3F
K= 10111111 10010001 10001101 00111101 00111111
----------F(R, K)----------
E(R)= E0 54 59 4A C0
E(R)= 11100000 01010100 01011001 01001010 11000000
E(R) xor K = 5F C5 D4 77 FF
E(R) xor K = 01011111 11000101 11010100 01110111 11111111
SBoxed= B2 E8 8D 3C
SBoxed= 10110010 11101000 10001101 00111100
P modification processed and result in F(R, K)...
---
F(R, K)= 5B 81 27 6E
F(R, K)= 01011011 10000001 00100111 01101110
R1= 43 42 32 34
R1= 01000011 01000010 00110010 00110100
L1= C2 8C 96 0D
L1= 11000010 10001100 10010110 00001101

---------- **Round 16** ----------
L= C2 8C 96 0D
L= 11000010 10001100 10010110 00001101
R= 43 42 32 34
R= 01000011 01000010 00110010 00110100
K= CB 3D 8B 0E 17
K= 11001011 00111101 10001011 00001110 00010111
----------F(R, K)----------
E(R)= 20 6A 04 1A 41
E(R)= 00100000 01101010 00000100 00011010 01000001
E(R) xor K = EB 57 8F 14 56
E(R) xor K = 11101011 01010111 10001111 00010100 01010110
SBoxed= A7 83 24 29
SBoxed= 10100111 10000011 00100100 00101001
P modification processed and result in F(R, K)...
---
F(R, K)= C8 C0 4F 98
F(R, K)= 11001000 11000000 01001111 10011000
R1= 0A 4C D9 95
R1= 00001010 01001100 11011001 10010101
L1= 43 42 32 34
L1= 01000011 01000010 00110010 00110100
IP1 data = 85 E8 13 54 0F 0A B4 05
IP1 data = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101
---
**Out data** = 85 E8 13 54 0F 0A B4 05
**Out data** = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101

**ج) عمليات رمزگشايي الگوريتم DESبا داشتن داده هاي كدگذاري شده و كليد مربوطه :**

عمليات Decryption معكوس عمليات كدگذاري است .
خلاصه به صورت pseudo code:

**Key schedule:**
C[0]D[0] = PC1(key)
for 1 <= i <= 16
C[i] = LS[i](C[i-1])
D[i] = LS[i](D[i-1])
K[i] = PC2(C[i]D[i])

**Encipherment:**
L[0]R[0] = IP(plain block)
for 1 <= i <= 16
L[i] = R[i-1]
R[i] = L[i-1] xor f(R[i-1], K[i])
cipher block = FP(R[16]L[16])

**Decipherment:**
R[16]L[16] = IP(cipher block)
for 1 <= i <= 16
R[i-1] = L[i]
L[i-1] = R[i] xor f(L[i], K[i])
plain block = FP(L[0]R[0])

و يا براي رمزگشايي همان پروسه ي اصلي بكار مي رود اما ساب كي ها به ترتيب عكس بكار گرفته مي شوند. يعني در ابتدا بجاي اعمال K1 ، K16 اعمال مي شود و به همين ترتيب.

**د) رمزگذاري داده هايي بزرگتر از ۶۴ بيت توسط الگوريتم DES**

در اين موارد بايد به روش هاي رسمي ذكر شده در FIPS PUB 81 مراجعه نمود (چهار روش ( ECB , CBC , OFB , CFB ) ذكر شده است).

**ه) Triple-DES**

همان DES است كه در آن دو كليد ۵۶ بيتي استفاده مي شود.

۱. ابتدا داده ها توسط الگوريتم نرمال DESتوسط كليد اول رمزگذاري مي شوند.
۲. سپس اين داده ي رمزگذاري شده با كليد دوم رمزگشايي مي شود.

چون كليد دوم كليد واقعي نبوده ، بنابراين اينكار سبب رمزگذاري بيشتر پيغام مي گردد.

۳. در آخر اين پيغامي كه دوبار رمزگذاري شده ، بار ديگر توسط كليد اول رمزگذاري مي شود تا پيغام نهايي تريپل-دس حاصل شود.

حالت هاي ديگري هم براي اين الگوريتم وجود دارد. براي مثال استفاده از سه كليد بجاي دو كليد و يا استفاده از دو كليد با ترتيب هايي ويژه در بكار گيري.

**و) شكستن الگوريتم DES !**

براي حالت brute force attack (امتحان كردن تك تك كليدهاي ممكن) ، ۲ به توان ۵۶ حالت وجود خواهد داشت! يعني حدود ۷۰،۰۰۰،۰۰۰،۰۰۰،۰۰۰ حالت!
براي مشاهده ي استحكام اين الگوريتم در حالت هاي مختلف مي توان به FIPBS PUB 74 مراجعه كرد:
http://www.itl.nist.gov/fipspubs/fip74.htm

گاهي از اوقات در دنيا مسابقه هايي رسمي براي شكستن اين الگوريتم ترتيب داده مي شود! براي مثال در سال ۱۹۹۹ در مسابقه ي RSA DES Challenge III، با استفاده از ۱۰۰ هزار كامپيوتر در سراسر دنيا موفق به شكستن اين الگوريتم پس از ۲۲ ساعت و ۱۵ دقيقه شدند!! (يعني توانايي تست كردن 245 billion keys per second)

وحيد نصيري
۱۳۸۳/۵/۱۸