Know plaintext cryptanalysis challenge
Difficulty : intermediate

1 Introduction

The goal is to break the cipher and recover the 64 bit key. To achieve this goal, the complete description of the cipher is provided, as well as some test vectors to make sure your implementation is correct and a set of pairs 300'000 of randomly generated plaintext/ciphertext, all encrypted with the secret key to recover.

Any detailed description of how and why the cipher was constructed this way, as well as how the s-boxes were generated are voluntary omitted. Obviously they are bad (except for the last one), and were designed to make sure the cipher can be broken with a limited amount of data and low complexity.

2 The Cipher

2.1 General Information

- **Type :** SPN
- **Blocksize :** 64 bits
- **Rounds :** 5
- **Subkeys :** 6
- **Key addition :** $K()$
- **Nonlinear layer :** $S()$
- **Linear layer :** $P()$
- **Key schedule :** *key_schedule*()

Each block should be viewed as an a tuple of 8 byte sized words : state = $(w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7)$.

The encryption scheme works as follow :

```
ENCRYPT(state,key):
  BEGIN
    keys = k_schedule(key)
    for i = 0 to 4 do:
      state = K(state,keys[i])
      state = S(state)
      state = P(state)
    end
    state = K(state,keys[i+1])
    state = S(state)
    state = K(state,keys[i+2])
    return state
  END
```

2.2 Key addition : $K()$

Given a subkey k_i of 64 bits, k_i is treated as tuple of 8 byte sized words :

$$k_i = (k_{i,0}, k_{i,1}, k_{i,2}, k_{i,3}, k_{i,4}, k_{i,5}, k_{i,6}, k_{i,7}).$$

Given a state $= (w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7)$,

$$K(\text{state}, k_i) = (z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7),$$

where

$$\begin{aligned} z_0 &= w_0 \oplus k_{i,0} \\ z_1 &= w_1 \oplus k_{i,1} \\ z_2 &= w_2 \oplus k_{i,2} \\ z_3 &= w_3 \oplus k_{i,3} \\ z_4 &= w_4 \oplus k_{i,4} \\ z_5 &= w_5 \oplus k_{i,5} \\ z_6 &= w_6 \oplus k_{i,6} \\ z_7 &= w_7 \oplus k_{i,7} \end{aligned}$$

2.3 Non linear layer : $S()$

The non linear part of the cipher is a bijective mapping which makes use of 5 different s-boxes ($S0$,$S1$,$S2$,$S3$,$S4$), mapping a set of 8 byte-words to an other set of 8 byte-words :

$$S() : \{0,1\}^8_0 \times \cdots \times \{0,1\}^8_7 \mapsto \{0,1\}^8_0 \times \cdots \times \{0,1\}^8_7.$$

Given a state $= (w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7)$:

$$S(\text{state}) = (z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7)$$

where for the first 4 rounds and the key schedule

$$\begin{aligned} z_0 &= S_0[w_0] \\ z_1 &= S_1[w_1] \\ z_2 &= S_2[w_2] \\ z_3 &= S_3[w_3] \\ z_4 &= S_1[w_4] \\ z_5 &= S_2[w_5] \\ z_6 &= S_3[w_6] \\ z_7 &= S_0[w_7] \end{aligned}$$

except for the last round where

$$\begin{aligned} z_0 &= S_4[w_0] \\ z_1 &= S_4[w_1] \\ z_2 &= S_4[w_2] \\ z_3 &= S_4[w_3] \\ z_4 &= S_4[w_4] \\ z_5 &= S_4[w_5] \\ z_6 &= S_4[w_6] \\ z_7 &= S_4[w_7] \end{aligned}$$

2.4 Linear layer : $P()$

The linear part of the cipher is a involutive bijective mapping :

$$P() = P()^{-1} : \{0,1\}^{64} \mapsto \{0,1\}^{64}.$$

Given a state $= (w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7)$:

$$P(\text{state}) = (z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7)$$

where

$$\begin{aligned}
z_0 &= w_2 \oplus w_3 \oplus w_4 \oplus w_6 \oplus w_7 \\
z_1 &= w_0 \oplus w_1 \oplus w_3 \oplus w_4 \oplus w_7 \\
z_2 &= w_0 \oplus w_1 \oplus w_4 \oplus w_5 \oplus w_6 \\
z_3 &= w_1 \oplus w_2 \oplus w_3 \oplus w_5 \oplus w_6 \\
z_4 &= w_0 \oplus w_2 \oplus w_3 \oplus w_6 \oplus w_7 \\
z_5 &= w_0 \oplus w_3 \oplus w_4 \oplus w_5 \oplus w_7 \\
z_6 &= w_0 \oplus w_1 \oplus w_2 \oplus w_4 \oplus w_5 \\
z_7 &= w_1 \oplus w_2 \oplus w_5 \oplus w_6 \oplus w_7
\end{aligned}$$

2.5 Key schedule : $key\_schedule()$

Given a key $= k$ of 64 bits, the key is treated as a tuple of 8 byte sized words :

$$k = (k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7)$$

Each subsequent key is computed from the previous one using a combination of the the previously defined functions :

```
key_schedule(k):
    BEGIN
        keys = [k]
        for i = 0 to 5 do:
            tmp = keys[i]
            tmp = $S$(tmp)
            tmp = $P$(tmp)
            tmp = $S$(tmp)
            keys = keys + tmp
        end
        return keys
    END
```

3 S-boxes

3.1 S0

```
0x8e ,0xb1 ,0x2c ,0xb3 ,0x8c ,0x11 ,0x2e ,0x13 ,0xf0 ,0x4f ,0x72 ,0x4d ,0xf2 ,0xcf ,0x70 ,0xcd ,
0xbe ,0x81 ,0x14 ,0x8b ,0xb4 ,0x21 ,0x1e ,0x2b ,0x40 ,0xff ,0xca ,0xf5 ,0x4a ,0x7f ,0xc0 ,0x75 ,
0xdf ,0x60 ,0x55 ,0x6a ,0xd5 ,0xe0 ,0x5f ,0xea ,0x01 ,0x3e ,0xab ,0x34 ,0x0b ,0x9e ,0xa1 ,0x94 ,
0x6f ,0xd0 ,0xed ,0xd2 ,0x6d ,0x50 ,0xef ,0x52 ,0x31 ,0x0e ,0x93 ,0x0c ,0x33 ,0xae ,0x91 ,0xac ,
0x1f ,0x20 ,0xb5 ,0x2a ,0x15 ,0x80 ,0xbf ,0x8a ,0x5e ,0xe1 ,0xd4 ,0xeb ,0x54 ,0x61 ,0xde ,0x6b ,
0xee ,0x51 ,0x6c ,0x53 ,0xec ,0xd1 ,0x6e ,0xd3 ,0x2f ,0x10 ,0x8d ,0x12 ,0x2d ,0xb0 ,0x8f ,0xb2 ,
0x71 ,0xce ,0xf3 ,0xcc ,0x73 ,0x4e ,0xf1 ,0x4c ,0x90 ,0xaf ,0x32 ,0xad ,0x92 ,0x0f ,0x30 ,0x0d ,
0xa0 ,0x9f ,0x0a ,0x95 ,0xaa ,0x3f ,0x00 ,0x35 ,0xc1 ,0x7e ,0x4b ,0x74 ,0xcb ,0xfe ,0x41 ,0xf4 ,
0x88 ,0xb7 ,0x22 ,0xbd ,0x82 ,0x17 ,0x28 ,0x1d ,0x69 ,0xd6 ,0xe3 ,0xdc ,0x63 ,0x56 ,0xe9 ,0x5c ,
0xdb ,0x64 ,0x59 ,0x66 ,0xd9 ,0xe4 ,0x5b ,0xe6 ,0xba ,0x85 ,0x18 ,0x87 ,0xb8 ,0x25 ,0x1a ,0x27 ,
0x44 ,0xfb ,0xc6 ,0xf9 ,0x46 ,0x7b ,0xc4 ,0x79 ,0x05 ,0x3a ,0xa7 ,0x38 ,0x07 ,0x9a ,0xa5 ,0x98 ,
0x37 ,0x08 ,0x9d ,0x02 ,0x3d ,0xa8 ,0x97 ,0xa2 ,0xf6 ,0x49 ,0x7c ,0x43 ,0xfc ,0xc9 ,0x76 ,0xc3 ,
0x1b ,0x24 ,0xb9 ,0x26 ,0x19 ,0x84 ,0xbb ,0x86 ,0xc5 ,0x7a ,0x47 ,0x78 ,0xc7 ,0xfa ,0x45 ,0xf8 ,
0x29 ,0x16 ,0x83 ,0x1c ,0x23 ,0xb6 ,0x89 ,0xbc ,0x77 ,0xc8 ,0xfd ,0xc2 ,0x7d ,0x48 ,0xf7 ,0x42 ,
0xe8 ,0x57 ,0x62 ,0x5d ,0xe2 ,0xd7 ,0x68 ,0xdd ,0x96 ,0xa9 ,0x3c ,0xa3 ,0x9c ,0x09 ,0x36 ,0x03 ,
0x5a ,0xe5 ,0xd8 ,0xe7 ,0x58 ,0x65 ,0xda ,0x67 ,0xa4 ,0x9b ,0x06 ,0x99 ,0xa6 ,0x3b ,0x04 ,0x39 ,
```

3.2 S1

```
0x5f ,0xe0 ,0x55 ,0x6a ,0xdf ,0xea ,0xd5 ,0x60 ,0xbe ,0x81 ,0xb4 ,0x2b ,0x1e ,0x8b ,0x14 ,0x21 ,
0x8c ,0xb3 ,0x8e ,0x11 ,0x2c ,0xb1 ,0x2e ,0x13 ,0xed ,0x52 ,0xef ,0xd0 ,0x6d ,0x50 ,0x6f ,0xd2 ,
0x33 ,0x0c ,0x31 ,0xae ,0x93 ,0x0e ,0x91 ,0xac ,0x72 ,0xcd ,0x70 ,0x4f ,0xf2 ,0xcf ,0xf0 ,0x4d ,
0xc0 ,0x7f ,0xca ,0xf5 ,0x40 ,0x75 ,0x4a ,0xff ,0x01 ,0x3e ,0x0b ,0x94 ,0xa1 ,0x34 ,0xab ,0x9e ,
0x92 ,0xad ,0x90 ,0x0f ,0x32 ,0xaf ,0x30 ,0x0d ,0x6c ,0xd3 ,0x6e ,0x51 ,0xec ,0xd1 ,0xee ,0x53 ,
0xa0 ,0x9f ,0xaa ,0x35 ,0x00 ,0x95 ,0x0a ,0x3f ,0xde ,0x61 ,0xd4 ,0xeb ,0x5e ,0x6b ,0x54 ,0xe1 ,
0x41 ,0xfe ,0x4b ,0x74 ,0xc1 ,0xf4 ,0xcb ,0x7e ,0x1f ,0x20 ,0x15 ,0x8a ,0xbf ,0x2a ,0xb5 ,0x80 ,
0xf3 ,0x4c ,0xf1 ,0xce ,0x73 ,0x4e ,0x71 ,0xcc ,0x2d ,0x12 ,0x2f ,0xb0 ,0x8d ,0x10 ,0x8f ,0xb2 ,
0x07 ,0x38 ,0x05 ,0x9a ,0xa7 ,0x3a ,0xa5 ,0x98 ,0x59 ,0xe6 ,0x5b ,0x64 ,0xd9 ,0xe4 ,0xdb ,0x66 ,
0x37 ,0x08 ,0x3d ,0xa2 ,0x97 ,0x02 ,0x9d ,0xa8 ,0xe9 ,0x56 ,0xe3 ,0xdc ,0x69 ,0x5c ,0x63 ,0xd6 ,
0x76 ,0xc9 ,0x7c ,0x43 ,0xf6 ,0xc3 ,0xfc ,0x49 ,0x88 ,0xb7 ,0x82 ,0x1d ,0x28 ,0xbd ,0x22 ,0x17 ,
0xc6 ,0x79 ,0xc4 ,0xfb ,0x46 ,0x7b ,0x44 ,0xf9 ,0xb8 ,0x87 ,0xba ,0x25 ,0x18 ,0x85 ,0x1a ,0x27 ,
0x68 ,0xd7 ,0x62 ,0x5d ,0xe8 ,0xdd ,0xe2 ,0x57 ,0x29 ,0x16 ,0x23 ,0xbc ,0x89 ,0x1c ,0x83 ,0xb6 ,
0x19 ,0x26 ,0x1b ,0x84 ,0xb9 ,0x24 ,0xbb ,0x86 ,0xd8 ,0x67 ,0xda ,0xe5 ,0x58 ,0x65 ,0x5a ,0xe7 ,
0xa6 ,0x99 ,0xa4 ,0x3b ,0x06 ,0x9b ,0x04 ,0x39 ,0x47 ,0xf8 ,0x45 ,0x7a ,0xc7 ,0xfa ,0xc5 ,0x78 ,
0xf7 ,0x48 ,0xfd ,0xc2 ,0x77 ,0x42 ,0x7d ,0xc8 ,0x96 ,0xa9 ,0x9c ,0x03 ,0x36 ,0xa3 ,0x3c ,0x09 ,
```

3.3 S2

```
0x29 ,0xb6 ,0x23 ,0x1c ,0x89 ,0xbc ,0x83 ,0x16 ,0xe8 ,0xd7 ,0xe2 ,0x5d ,0x68 ,0xdd ,0x62 ,0x57 ,
0x5a ,0x65 ,0x58 ,0xe7 ,0xda ,0x67 ,0xd8 ,0xe5 ,0x1b ,0x84 ,0x19 ,0x26 ,0xbb ,0x86 ,0xb9 ,0x24 ,
0xc5 ,0xfa ,0xc7 ,0x78 ,0x45 ,0xf8 ,0x47 ,0x7a ,0xa4 ,0x3b ,0xa6 ,0x99 ,0x04 ,0x39 ,0x06 ,0x9b ,
0x96 ,0x09 ,0x9c ,0xa3 ,0x36 ,0x03 ,0x3c ,0xa9 ,0x77 ,0x48 ,0x7d ,0xc2 ,0xf7 ,0x42 ,0xfd ,0xc8 ,
0xdb ,0xe4 ,0xd9 ,0x66 ,0x5b ,0xe6 ,0x59 ,0x64 ,0x05 ,0x9a ,0x07 ,0x38 ,0xa5 ,0x98 ,0xa7 ,0x3a ,
0x69 ,0x56 ,0x63 ,0xdc ,0xe9 ,0x5c ,0xe3 ,0xd6 ,0x37 ,0xa8 ,0x3d ,0x02 ,0x97 ,0xa2 ,0x9d ,0x08 ,
0x88 ,0x17 ,0x82 ,0xbd ,0x28 ,0x1d ,0x22 ,0xb7 ,0xf6 ,0xc9 ,0xfc ,0x43 ,0x76 ,0xc3 ,0x7c ,0x49 ,
0xba ,0x25 ,0xb8 ,0x87 ,0x1a ,0x27 ,0x18 ,0x85 ,0x44 ,0x7b ,0x46 ,0xf9 ,0xc4 ,0x79 ,0xc6 ,0xfb ,
0xee ,0xd1 ,0xec ,0x53 ,0x6e ,0xd3 ,0x6c ,0x51 ,0x90 ,0x0f ,0x92 ,0xad ,0x30 ,0x0d ,0x32 ,0xaf ,
0x5e ,0x61 ,0x54 ,0xeb ,0xde ,0x6b ,0xd4 ,0xe1 ,0xa0 ,0x3f ,0xaa ,0x95 ,0x00 ,0x35 ,0x0a ,0x9f ,
0x1f ,0x80 ,0x15 ,0x2a ,0xbf ,0x8a ,0xb5 ,0x20 ,0xc1 ,0xfe ,0xcb ,0x74 ,0x41 ,0xf4 ,0x4b ,0x7e ,
0x2f ,0xb0 ,0x2d ,0x12 ,0x8f ,0xb2 ,0x8d ,0x10 ,0x71 ,0x4e ,0x73 ,0xcc ,0xf1 ,0x4c ,0xf3 ,0xce ,
0xbe ,0x21 ,0xb4 ,0x8b ,0x1e ,0x2b ,0x14 ,0x81 ,0xdf ,0xe0 ,0xd5 ,0x6a ,0x5f ,0xea ,0x55 ,0x60 ,
0x6f ,0x50 ,0x6d ,0xd2 ,0xef ,0x52 ,0xed ,0xd0 ,0x8e ,0x11 ,0x8c ,0xb3 ,0x2e ,0x13 ,0x2c ,0xb1 ,
0xf0 ,0xcf ,0xf2 ,0x4d ,0x70 ,0xcd ,0x72 ,0x4f ,0x31 ,0xae ,0x33 ,0x0c ,0x91 ,0xac ,0x93 ,0x0e ,
0x01 ,0x9e ,0x0b ,0x34 ,0xa1 ,0x94 ,0xab ,0x3e ,0x40 ,0x7f ,0x4a ,0xf5 ,0xc0 ,0x75 ,0xca ,0xff ,
```

3.4 S3

```
0xd2 ,0x6f ,0x50 ,0x6d ,0x52 ,0xed ,0xd0 ,0xef ,0x8c ,0xb1 ,0x2e ,0xb3 ,0x2c ,0x13 ,0x8e ,0x11 ,
0x1c ,0x29 ,0xb6 ,0x23 ,0xbc ,0x83 ,0x16 ,0x89 ,0xe2 ,0x57 ,0x68 ,0x5d ,0x62 ,0xdd ,0xe8 ,0xd7 ,
0xa3 ,0x96 ,0x09 ,0x9c ,0x03 ,0x3c ,0xa9 ,0x36 ,0x7d ,0xc8 ,0xf7 ,0xc2 ,0xfd ,0x42 ,0x77 ,0x48 ,
0x4d ,0xf0 ,0xcf ,0xf2 ,0xcd ,0x72 ,0x4f ,0x70 ,0x33 ,0x0e ,0x91 ,0x0c ,0x93 ,0xac ,0x31 ,0xae ,
0x63 ,0xd6 ,0xe9 ,0xdc ,0xe3 ,0x5c ,0x69 ,0x56 ,0x02 ,0x37 ,0xa8 ,0x3d ,0xa2 ,0x9d ,0x08 ,0x97 ,
0x53 ,0xee ,0xd1 ,0xec ,0xd3 ,0x6c ,0x51 ,0x6e ,0x92 ,0xaf ,0x30 ,0xad ,0x32 ,0x0d ,0x90 ,0x0f ,
0x2d ,0x10 ,0x8f ,0x12 ,0x8d ,0xb2 ,0x2f ,0xb0 ,0xcc ,0x71 ,0x4e ,0x73 ,0x4c ,0xf3 ,0xce ,0xf1 ,
0xbd ,0x88 ,0x17 ,0x82 ,0x1d ,0x22 ,0xb7 ,0x28 ,0xfc ,0x49 ,0x76 ,0x43 ,0x7c ,0xc3 ,0xf6 ,0xc9 ,
0xde ,0x6b ,0x54 ,0x61 ,0x5e ,0xe1 ,0xd4 ,0xeb ,0x3f ,0x0a ,0x95 ,0x00 ,0x9f ,0xa0 ,0x35 ,0xaa ,
0xe4 ,0x59 ,0x66 ,0x5b ,0x64 ,0xdb ,0xe6 ,0xd9 ,0xa5 ,0x98 ,0x07 ,0x9a ,0x05 ,0x3a ,0xa7 ,0x38 ,
0x1a ,0x27 ,0xb8 ,0x25 ,0xba ,0x85 ,0x18 ,0x87 ,0x7b ,0xc6 ,0xf9 ,0xc4 ,0xfb ,0x44 ,0x79 ,0x46 ,
0x80 ,0xb5 ,0x2a ,0xbf ,0x20 ,0x1f ,0x8a ,0x15 ,0x41 ,0xf4 ,0xcb ,0xfe ,0xc1 ,0x7e ,0x4b ,0x74 ,
0x65 ,0xd8 ,0xe7 ,0xda ,0xe5 ,0x5a ,0x67 ,0x58 ,0xbb ,0x86 ,0x19 ,0x84 ,0x1b ,0x24 ,0xb9 ,0x26 ,
0x21 ,0x14 ,0x8b ,0x1e ,0x81 ,0xbe ,0x2b ,0xb4 ,0x5f ,0xea ,0xd5 ,0xe0 ,0xdf ,0x60 ,0x55 ,0x6a ,
0x9e ,0xab ,0x34 ,0xa1 ,0x3e ,0x01 ,0x94 ,0x0b ,0xc0 ,0x75 ,0x4a ,0x7f ,0x40 ,0xff ,0xca ,0xf5 ,
0xfa ,0x47 ,0x78 ,0x45 ,0x7a ,0xc5 ,0xf8 ,0xc7 ,0x04 ,0x39 ,0xa6 ,0x3b ,0xa4 ,0x9b ,0x06 ,0x99 ,
```

3.5 S4

```
0x3f,0xd6,0x16,0x1a,0x1d,0x14,0xe2,0xd0,0x3c,0x00,0xb0,0xe7,0x90,0xa7,0xb6,0x1f,
0x60,0xe1,0xd2,0xc6,0x6b,0xba,0x25,0xbe,0xf1,0x56,0xce,0x2d,0x94,0x0c,0x78,0x9c,
0x6d,0x30,0x0d,0x11,0x17,0x80,0x71,0x1e,0x7c,0xac,0x8b,0xaa,0xe4,0xe5,0xb5,0xea,
0x83,0x9a,0x03,0xc3,0x42,0x10,0xab,0xd8,0x27,0xec,0x09,0x12,0x62,0x92,0xb4,0x43,
0xe9,0x5b,0xdf,0xc2,0x4a,0x3b,0xe8,0x4b,0x08,0xd4,0x45,0x32,0x95,0x97,0x98,0xbc,
0x77,0x86,0x7d,0x85,0x79,0x02,0x4d,0x73,0x5a,0x4f,0xcf,0xca,0xa1,0x6e,0x50,0xad,
0x7b,0x1c,0xff,0x5f,0x19,0x01,0x46,0x63,0x5c,0xda,0xa8,0x6c,0x3a,0x57,0x7a,0x4c,
0x65,0xef,0x04,0xe6,0xfb,0xf0,0x35,0xb2,0x06,0xd5,0x87,0x61,0x59,0xf6,0x9f,0xb9,
0x44,0xae,0xd9,0xa0,0x55,0x8a,0x48,0xa9,0x20,0xc5,0x91,0x0f,0xa3,0xb1,0x0b,0x8f,
0x0a,0xc7,0x9b,0x41,0xeb,0x54,0xf8,0x18,0xdd,0x6a,0x72,0xc4,0xc1,0x2e,0xf4,0xa5,
0xb7,0x37,0xbb,0x29,0xc0,0x82,0x52,0x6f,0x7f,0xa2,0xcc,0x39,0xde,0xfe,0x8e,0x2a,
0x26,0xe0,0x53,0x05,0x67,0x99,0x9d,0x49,0x34,0x4e,0xdb,0x76,0xe3,0x21,0x13,0x1b,
0x36,0x07,0xcd,0xa4,0xc8,0x3e,0xd7,0xb3,0x74,0x38,0x8c,0x40,0xf5,0x96,0x89,0x2b,
0x47,0x22,0x24,0xb8,0xf3,0x8d,0x9e,0xcb,0xbd,0x5e,0x84,0xaf,0x93,0xa6,0xf9,0x75,
0x66,0x28,0x88,0x15,0xed,0xee,0x51,0xdc,0x0e,0xf2,0x58,0x2c,0x64,0x23,0x70,0xd1,
0xfc,0x69,0xfd,0x81,0x2f,0xbf,0x3d,0xf7,0x5d,0x31,0x33,0xfa,0x68,0x7e,0xd3,0xc9,
```

4 Test vectors

Encryption of the plaintext 0x0000000000000000 with the key 0x0123456789abcdef :

```
key 0       : 01 23 45 67 89 ab cd ef -> subkey 0 : 01 23 45 67 89 ab cd ef

key 1, S()  : b1 ae e6 b0 e6 74 24 03
key 1, P()  : 97 4a a9 a8 c0 90 6b 1b
key 1, S()  : e6 6e fe 7b 68 5e 73 f5 -> subkey 1 : e6 6e fe 7b 68 5e 73 f5

key 2, S()  : 68 b5 ca 43 1f 9d 82 65
key 2, P()  : 71 e4 dd 23 06 cc 95 05
key 2, S()  : 9f 06 13 9c d5 5f db 11 -> subkey 2 : 9f 06 13 9c d5 5f db 11

key 3, S()  : 27 d5 e7 05 24 08 e0 81
key 3, P()  : a7 52 3e df a4 8f 39 5b
key 3, S()  : 79 aa fd 6a f6 af 0e 12 -> subkey 3 : 79 aa fd 6a f6 af 0e 12

key 4, S()  : 7e 82 75 4e 7d 7e 8e 14
key 4, P()  : dc db 71 49 df 27 8a 13
key 4, S()  : 7d e5 25 37 e7 7a 95 8b -> subkey 4 : 7d e5 25 37 e7 7a 95 8b

key 5, S()  : fe 9b f8 70 39 46 db dc
key 5, P()  : b6 f0 c1 8e 71 2d e2 22
key 5, S()  : 97 f7 21 35 4c 39 34 55 -> subkey 5 : 97 f7 21 35 4c 39 34 55
```

```
plaintext :     : 00 00 00 00 00 00 00 00

Round 1, state  : 00 00 00 00 00 00 00 00
Round 1, K()    : 01 23 45 67 89 ab cd ef
Round 1, S()    : b1 ae e6 b0 e6 74 24 03
Round 1, P()    : 97 4a a9 a8 c0 90 6b 1b

Round 2, state  : 97 4a a9 a8 c0 90 6b 1b
Round 2, K()    : 71 24 57 d3 a8 ce 18 ee
Round 2, S()    : 9f 93 d6 1e 88 55 e2 36
Round 2, P()    : 94 ac 33 ec 83 6a 07 c4

Round 3, state  : 94 ac 33 ec 83 6a 07 c4
Round 3, K()    : 0b aa 20 70 56 35 dc d5
Round 3, S()    : 4d 82 c5 bd 0a 03 df b6
Round 3, P()    : 1b ce 19 26 5c 4f 03 2d

Round 4, state  : 1b ce 19 26 5c 4f 03 2d
Round 4, K()    : 62 64 e4 4c aa e0 0d 3f
Round 4, S()    : f3 c1 70 a2 82 f0 13 ac
Round 4, P()    : ef be 53 f0 9e 8f 30 fe

Round 5, state  : ef be 53 f0 9e 8f 30 fe
Round 5, K()    : 92 5b 76 c7 79 f5 a5 75
Round 5, S()    : 9b ca 35 b3 d5 bf 82 f0
Round 5, K()    : 0c 3d 14 86 99 86 b6 a5

ciphertext      : 0c 3d 14 86 99 86 b6 a5
```