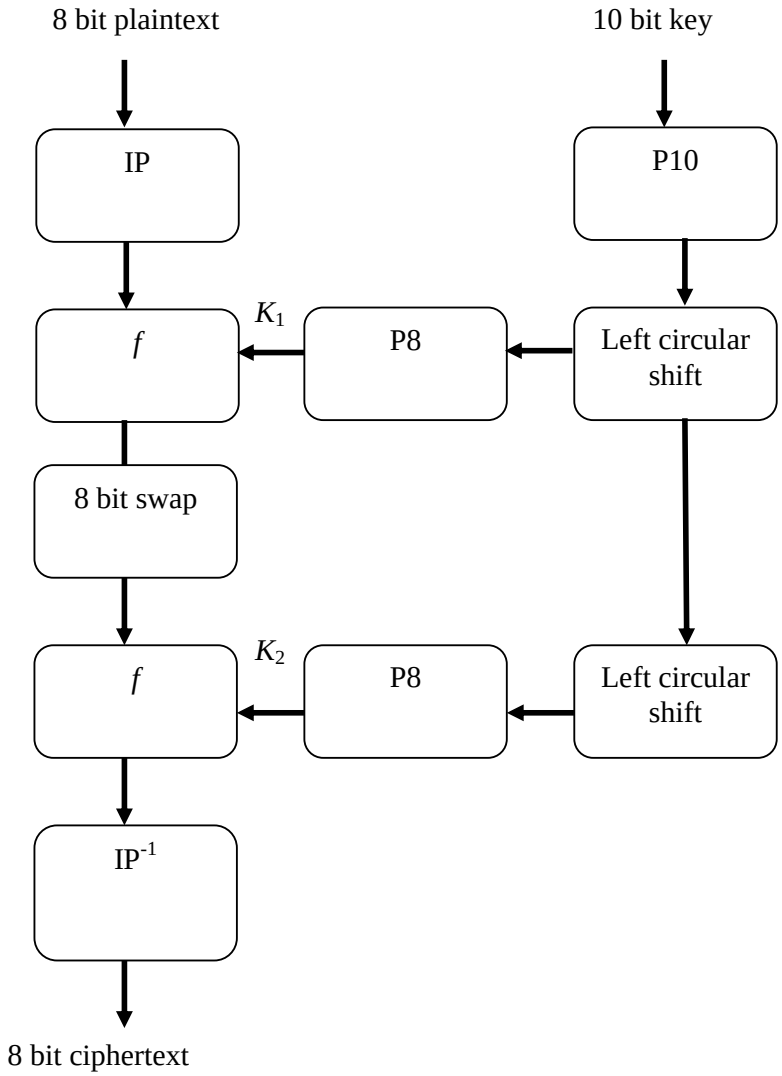
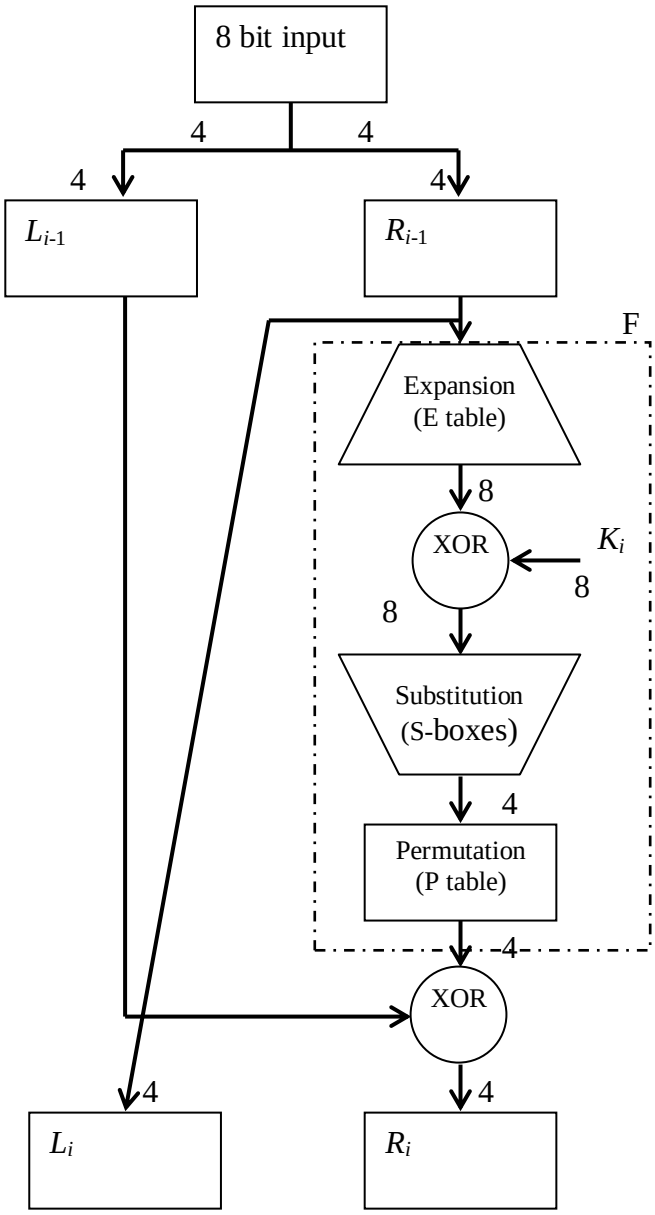
Simplified DES.

Developed by Prof. E. Schaefer this is a teaching tool and not a secure encryption algorithm. It uses a 10 bit key, an 8 bit plaintext and has only 2 iterations.



Overview



The function *f*

The associated tables:

P10

Input	1	2	3	4	5	6	7	8	9	10
output	3	5	2	7	4	10	1	9	8	6

Ex: input: 1111100000 output: 1110101000

P8

Input	1	2	3	4	5	6	7	8
output	6	3	7	4	8	5	10	9

Ex: input: 1111011110 output: 11111001
(using letters to name each bit)
Ex input: abcdefghij output: fcgdheji

IP

Input	1	2	3	4	5	6	7	8
output	2	6	3	1	4	8	5	7

Example input abcdefgh output: bfcadheg

IP^{-1}

Input	1	2	3	4	5	6	7	8
output	4	1	3	5	7	2	8	6

Example input bfcadheg output: abcdefgh

Expansion

Input	1	2	3	4	5	6	7	8
output	4	1	2	3	2	3	4	1

Example input abcd output: dabcbcda

The s-boxes:

S0	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2
S1				
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

Example: input 10010110 is split to form 1001 and 0110.
1001 is used with S0 and 0110 with S1.
With 1001 the first and last bits are put together to give the row, the middle bits are used to determine the column. Thus, row =11 = 3 column =0. In S0 row 3 column 0 the entry is 3=11 which is the output.

P table:

P

Input	1	2	3	4
output	2	4	3	1

Example: input: abcd output: bdca

Worked example:

Let's make up a key and a plaintext:
Plaintext: 10111101
Key: 1001001010

First we'll calculate k_1 and k_2 as these can be done separately.

k_1=P8(LCS(P10(key)))

P10(Key) = P10(1001001010) = 0001101100

LCS(P10(Key)) =LCS(00011 01100)= 00110 11000

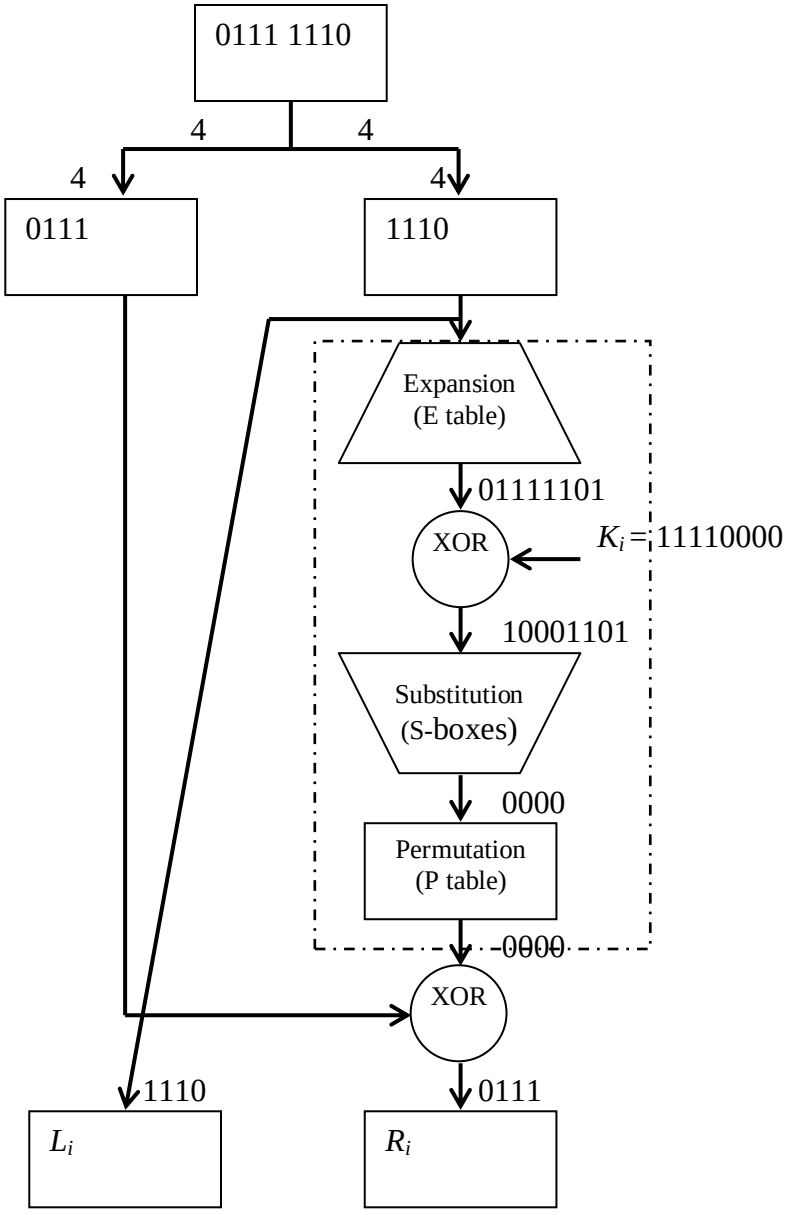
P8(00110 11000) = 11110000
$\underline{k_1=11110000}$

k_2=P8(LCS(LCS(P10(key))))

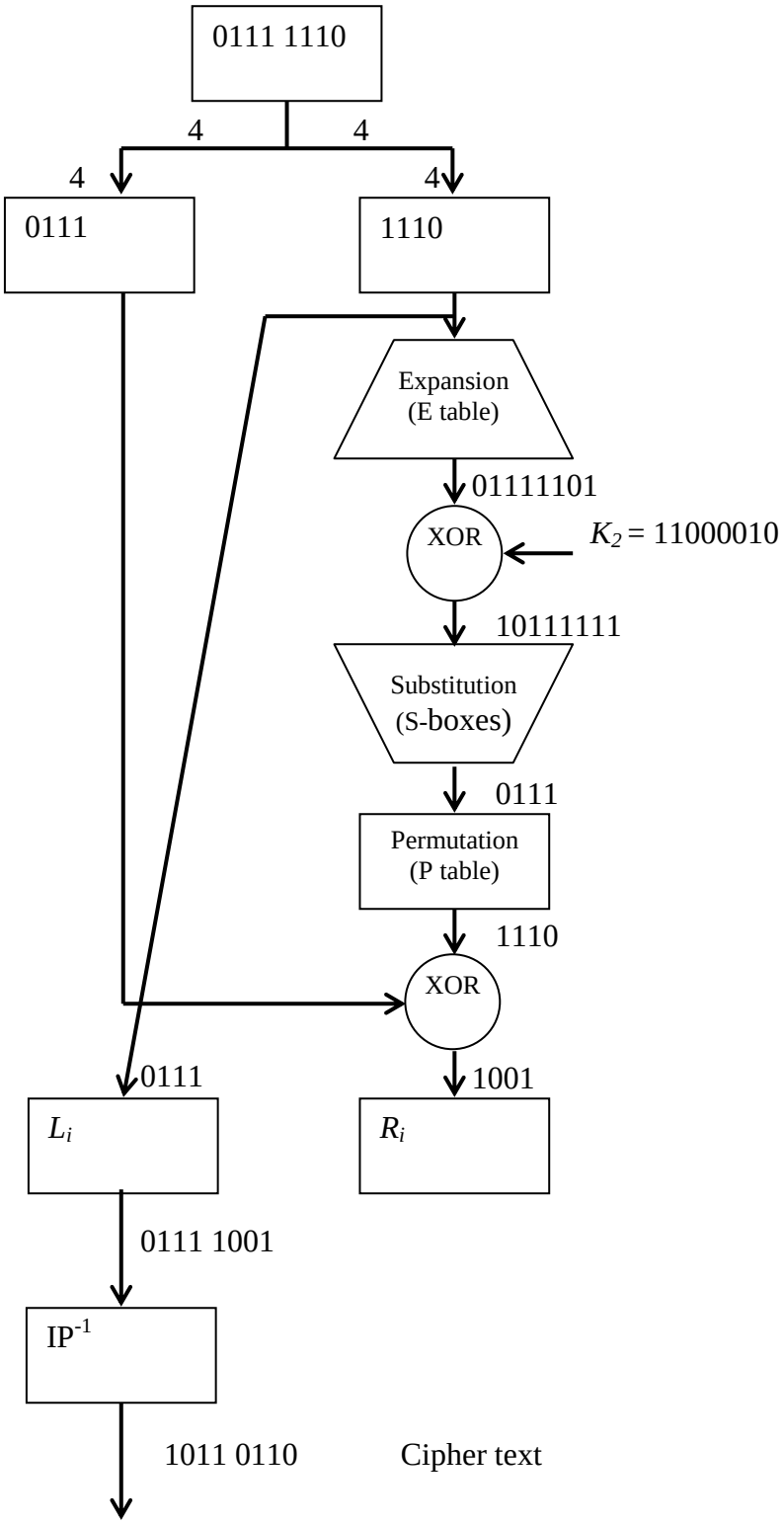
LCS(P10(key)) = 00110 11000
LCS(00110 11000) = 01100 10001
P8(01100 10001) = 11000010
$\underline{k_2=11000010}$

The 8-bit plaintext is then coded by IP as:
IP(p) = IP(1011 1101) =0111 1110

This enters *f* which is shown below:



Iteration 1.



Iteration 2 + IP^{-1} to give plaintext.

DES Weak Keys

- with many block ciphers there are some keys that should be avoided, because of reduced cipher complexity
- these keys are such that the same sub-key is generated in more than one round

Weak Keys:

  - The same sub-key is generated for every round
  - DES has 4 weak keys

Semi-Weak Keys

  - only two sub-keys are generated on alternate rounds
  - DES has 12 of these (in 6 pairs)

- **Complement Keys** k **and it's complement** k'
  - $E_k(P) = C$
  - $E_{k'}(P') = C'$
  - Results from the XOR operation of the keys with the expansion permutation.
  - Every sub-key will be the complement also.
  - This is not much of a problem if we're careful.

DES variations

Double DES:

- Use 2 keys: $k1$ and $k2$.
- Encryption is $E_{k2}(E_{k1}(P))$, decryption is simple: $D_{k1}(D_{k2}(P))$
- Increases the number of keys from 2^{56} to $2^{56\times2}$
- Note this would be a problem if there existed another key such that
- $E_{k1}(E_{k2}(P)) = E_{k3}(P)$ i.e. if DES was a group.
- Proved not to be true in 1992.

Meet in the middle attacks:
If $C=E_{k2}(E_{k1}(P))$ then
$X= E_{k1}(P) = D_{k2}(C)$

Now if we know a few plaintext-ciphertext pairs (P_1,C_1) then we can calculate all $E_k(P_1)$ and compare them to $D_k(C_1)$.

i.e. we encrypt for all possible values of P_1 and compare with decrypted values of C_1 using random keys. If there is a match then test using (P_2,C_2) if this is successful then we have the keys.

Attack time: 2^{64} possible ciphertexts, 2^{112} possible keys.

Therefore the number of keys that will produce a given ciphertext is $2^{112}/2^{64} = 2^{48}$.

However the chances of the keys encrypting (P_1,C_1) and (P_2,C_2) is 2^{48-64} thus the effort required for this attack is the time it takes to encrypt P_1 for all keys = 2^{56}.

Triple DES:

- Use 2 or 3 keys
- Encryption:
  - $E_{k1}(E_{k2}(E_{k3}(P)))$ Key size = 168bits.
  - $E_{k1}(D_{k2}(E_{k1}(P)))$ Adopted for several standards.
- Meet in the middle attack now takes in the order of 2^{112} operations.

Other ciphers of interest are: Blowfish, RC5, RC4 and AES