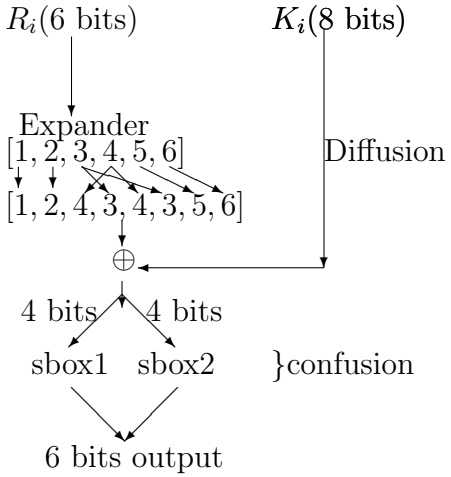
1. Simplified DES (SDES)

Feistel Cipher
· Block size: 12 bits
· Number of rounds 3 (or 4)
· The inner function ($f(R, k)$)

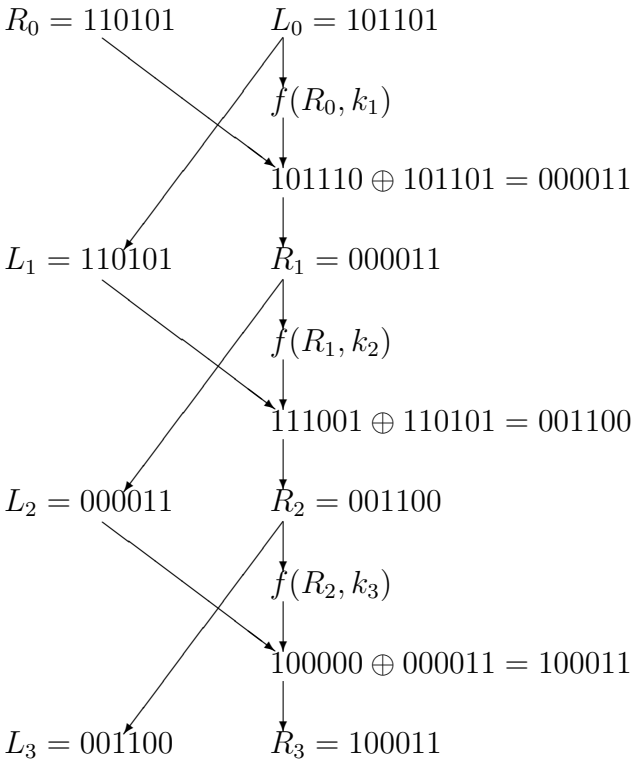


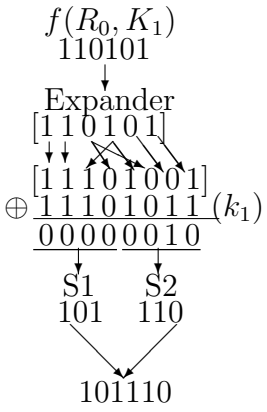
	0	1	2	3	4	5	6	7
S1	101	010	001	110	011	100	111	000
S1	001	100	110	010	000	111	101	011
S2	100	000	110	101	111	001	011	010
S2	101	011	000	111	110	010	001	100

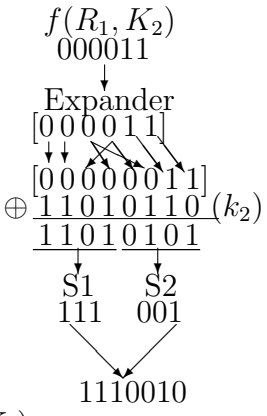
2 goals of symmetric crypto system (shannon)
Confusion- the relationship between the key and the ciphertext should be complicated that one cant easily find the key from CT
Diffusion-Ciphertext should depend on every bit of the plaintext
on average half the bits of CT change if one bit of plaintext changes

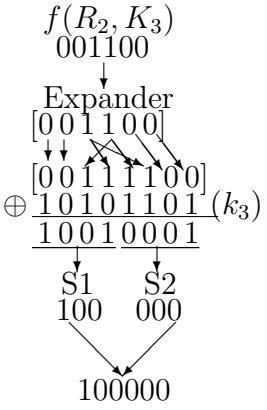
ith Round key K_i is going to be the 8 bits starting from the ith bit of the master and wrapping around

1.1. **Ex.** Encrypt plaintext
P=101101 110101
K=111 010 110
$k_1 = 11101011$ $k_2 = 11010110$ $k_3 = 10101101$



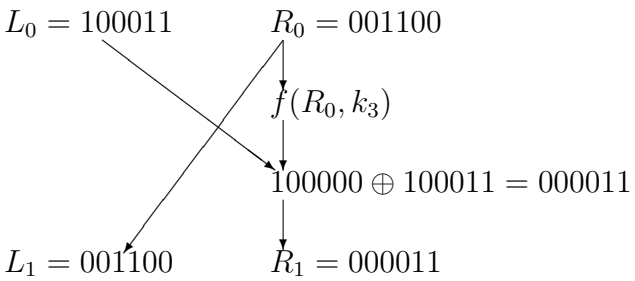






Ciphertext=001100100011

Decryption
swap left/right halves
c=001100 100011 $\rightarrow$ 100011 001100



2 more rounds of decryption
swap left/right to get back plain text

2. DES

Invented by IBM in 1972 called "LUCIFER"
64 bit plaintexts
56 bit master key
16 rounds
used from 1972-2000ish as the standard on the internet

How do we encrpyt Plaintexts longer than blocksize?
Break into "Chunks" P_1, P_2 easch has the size of one block
one idea: encrypt each block separately one after another
Electronic codebook (ECB)
$C_i = E_k(P_i)$

Cipher Block chaining
Pick one random C_0 Random string sent unencrypted
$C_i = E_k(P_i \oplus C_{i-1})$
to recover the plaintext
$P_i = D_k(C_i) \oplus C_{i-1}$

Methods for encrypting plaintexts are called Modes of operation

Electronic Codebook	(ECB)
Cipher Block Chaining	(CBC)
Propagating CBC	(PCBC)
Cipher Feedback	(CFB)
Output Feedback	(OFB)
Counter	(CTR)