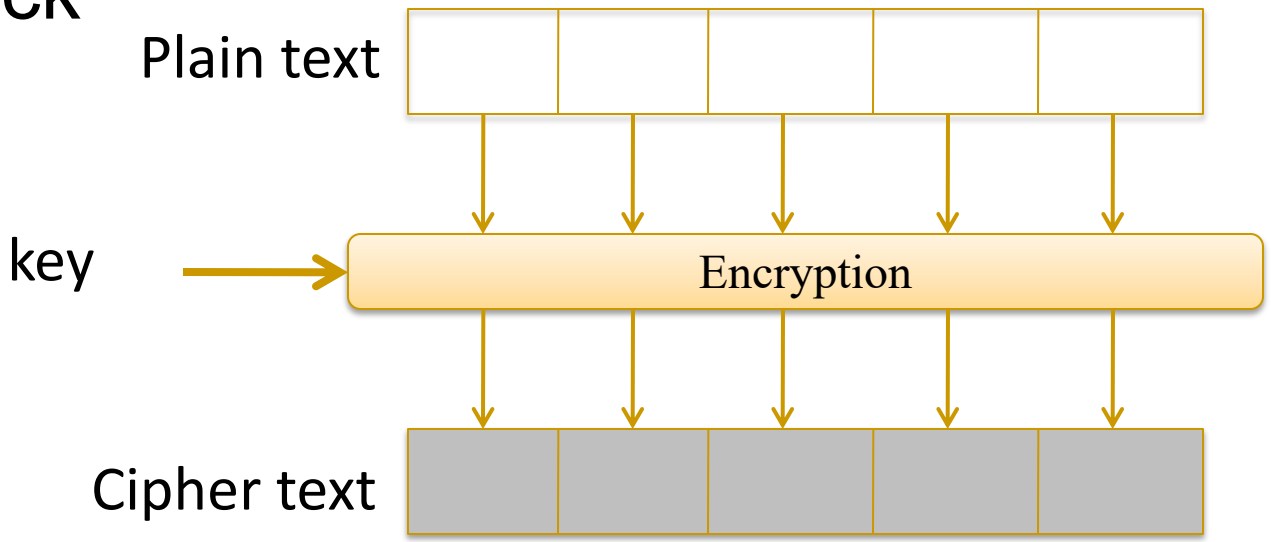
Cryptography II

Block ciphers and modes of operations

Block ciphers: getting the concept

- The input (binary bit string) is divided into blocks of fixed length
- The encryption and decryption are applied on each block



Block ciphers: getting the concept

key	000	001	010	011	100	101	110	111
0	001	111	110	000	100	010	101	011
1	001	110	111	100	011	010	000	101
2	001	000	100	101	110	111	010	011
3	100	101	110	111	000	001	010	011
4	101	110	100	010	011	001	011	111

- Plaintext= 010100110111= (010)(100)(110)(111)
  - ➔ Ciphertext = 111 011 000 101 key=1
  - ➔ Ciphertext = 100 011 011 111 key=4
- There are 5 keys, $2^2 < 5 < 2^3$ ➔ need keys in 3 bits to present➔ key size= block size= 3.
- Small sizes are dangerous, however: If Eve catches C=001 ➔ can infer P= 000 or 101.

General condition in creating secure block ciphers

- The block size has to be large enough to prevent against statistical analysis
  - However, larger block size means slower processing
- The key space (then key length) must be large enough to prevent against exhaustive key search
  - However, key length shouldn't be too big that makes key distribution and management more difficult

General principles in designing secure block ciphers

- *Confusion:* As a function, the dependence of the ciphertext on the plaintext should be complex enough so that enemy can't find the rules
  - The function should be non-linear.
- *Diffusion:* The goal is to spread the information from the plaintext over the entire ciphertext so that changes in plaintext affect many parts in ciphertext
  - This makes it difficult for an enemy to break the cipher by using statistical analysis
- Confusion is made by usings substitutions while *diffusion* by transpositions and/or permutations.

General principles in designing secure block ciphers

- Confusion
  - Plaintext : $T = t_1 t_2$
  - Cipher text : $M = m_1 m_2$

$$T' = t_1' t_2'$$
$$M' = m_1' m_2'$$

$$\begin{cases} m_1 = k_{11} t_1 + k_{12} t_2 \\ m_2 = k_{21} t_1 + k_{22} t_2 \end{cases} \qquad \begin{cases} m_1' = k_{11} t_1' + k_{12} t_2' \\ m_2' = k_{21} t_1' + k_{22} t_2' \end{cases}$$

$$\begin{cases} m_1 = k_{11} t_1 + k_{12} t_2 \\ m_1' = k_{11} t_1' + k_{12} t_2' \end{cases} \qquad \begin{cases} m_2 = k_{21} t_1 + k_{22} t_2 \\ m_2' = k_{21} t_1' + k_{22} t_2' \end{cases}$$

General principles in designing secure block ciphers

- Confusion
  - Solution: complex substitution

General principles in designing secure block ciphers

- Diffusion
  - Permutation + operation

c	o	m	p	u
t	e	r	s	e
c	u	r	i	t
y				

uetpsimrroeuctcy

computersecurity

xtgfncrjkthpucwx

Block cipher: main idea

- Block ciphers are usually designed with many rounds where basic round accomplishes the core function g for basic confusion and diffusion.
  - The input of a round is the output of the previous round and a subkey which is generated by a key-schedule algorithm
- The decryption is a reverse process where the sub-keys are handled in the reverse order

Substitution and Permutation network (SPN)

- Key schedule algorithm
  - Input: main key K
  - Output: N_r sub keys: $\{K^1, K^2, \ldots, K^{N_r}\}$
- Round function
  - $g(w^{r-1}, K^r)$

$$w^0 \leftarrow x$$
$$w^1 \leftarrow g(w^0, K^1)$$
$$w^2 \leftarrow g(w^1, K^2)$$
$$\ldots\ldots$$
$$w^{N_{r-1}} \leftarrow g(w^{N_{r-2}}, K^{N_{r-1}})$$
$$w^{N_r} \leftarrow g(w^{N_{r-1}}, K^{N_r})$$
$$y \leftarrow w^{N_r}$$

Substitution and Permutation network (SPN)

- Block length: $l \times m$
- Round function
  - Substitution: π_S
    - S-box: replacing l bits with different set of other l bits
  - Permutation: π_P
    - P-box: permuting lm bits

$$u^r \leftarrow w^{r-1} \oplus K^r$$
$$v^r \leftarrow \pi_S(u^r)$$
$$w^r \leftarrow \pi_P(v^r)$$

An example of round r

Substitution and Permutation network (SPN)

- An example: $SPN(x, \pi_S, \pi_P, (K^1, \ldots, K^{N_r+1}))$

$w^0 \leftarrow x$
For $r \leftarrow 1$ to $N_r - 1$
    $u^r \leftarrow w^{r-1} \oplus K^r$
    For $i \leftarrow 1$ to m // divide into m blocks, each of length l
        $v_i^r \leftarrow \pi_S(v_i^r)$ // substitution
    end
    $w^r \leftarrow \pi_P(v^r)$ // permutation
End
For $i \leftarrow 1$ to m
    $v_i^{N_r} \leftarrow \pi_S(v_i^{N_r})$
$y \leftarrow v_i^{N_r} \oplus K^{N_r+1}$

Substitution and Permutation network (SPN)

- Exercise: $l = m = N_r = 4$

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_P(z)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

K^1= 0011 1010 1001 0100
K^2= 1010 1001 0100 1101
K^3= 1001 0100 1101 0110
K^4= 0100 1101 0110 0011
K^5= 1101 0110 0011 1111

Plain text
$x = 0010\ 0110\ 1011\ 0111$

$w^0 \leftarrow x$
For $r \leftarrow 1$ to $N_r - 1$
  $u^r \leftarrow w^{r-1} \oplus K^r$
  For $i \leftarrow 1$ to m
    $\{u_1^r, \ldots, u_m^r\} \leftarrow u^r$ // divide into m blocks, each of length l
    $v_i^r \leftarrow \pi_S(u_i^r)$ // substitution
  end
  $v^r \leftarrow \{v_1^r, \ldots, v_m^r\}$ // concatenation
  $w^r \leftarrow \pi_P(v^r)$ // permutation
End
$u^{N_r} \leftarrow w^{N_r-1} \oplus K^{N_r}$
For $i \leftarrow 1$ to m
  $v_i^{N_r} \leftarrow \pi_S(u_i^{N_r})$
$y \leftarrow v^{N_r} \oplus K^{N_r+1}$

No permutation

No substitution and permutation

w^0	0010	0110	1011	0111
K^1	0011	1010	1001	0100
u^1	0001	1100	0010	0011
v^1	0100	0101	1101	0001
w^1	0010	1110	0000	0111

Substitution and Permutation network (SPN)

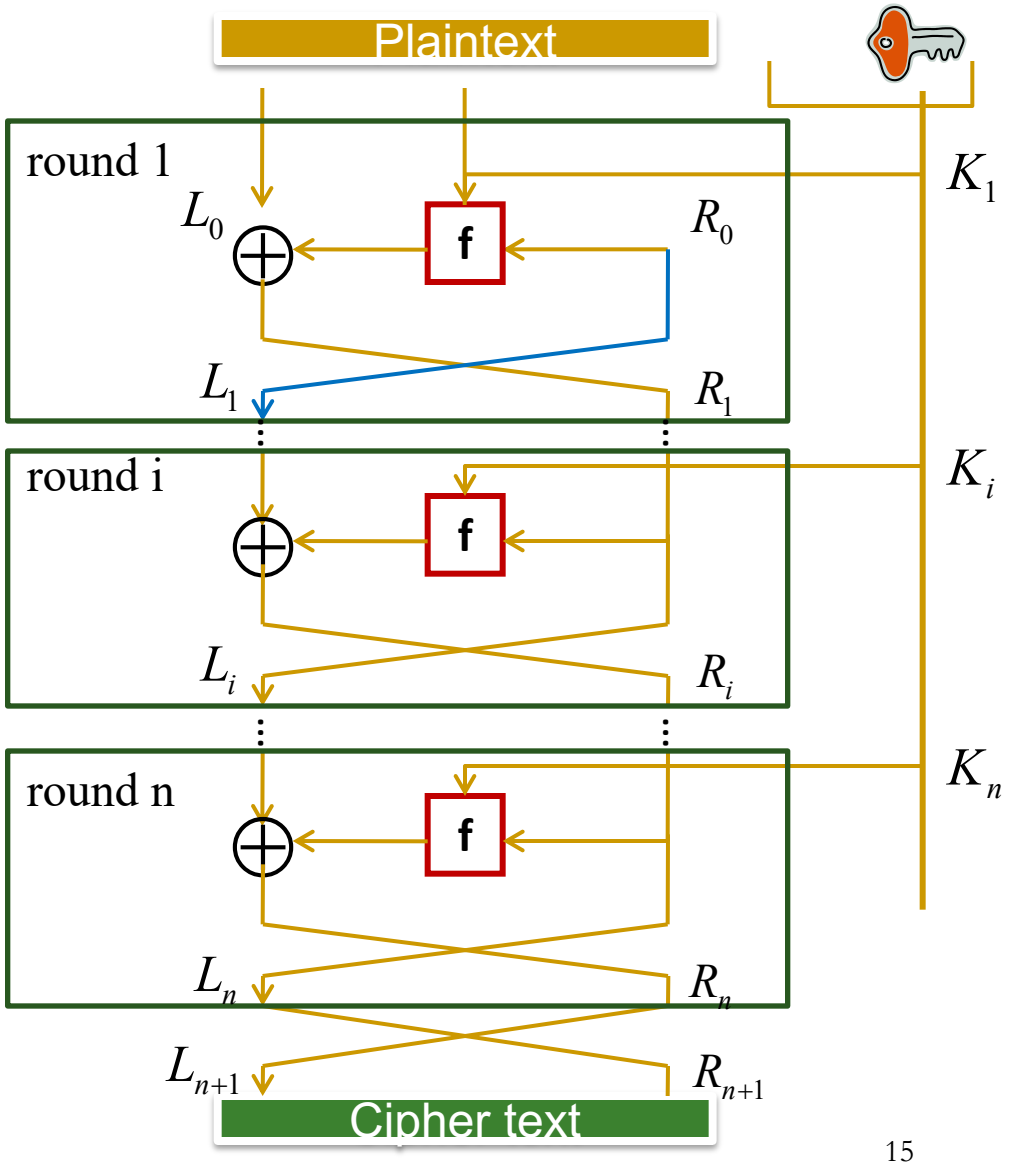
- Exercise: $l = m = N_r = 4$

w^0	0010	0110	1011	0111
K^1	0011	1010	1001	0100
u^1	0001	1100	0010	0011
v^1	0100	0101	1101	0001
w^1	0010	1110	0000	0111
K^2	1010	1001	0100	1101
u^2	1000	0111	0100	1010
v^2	0011	1000	0010	0110
w^2	0100	0001	1011	1000

K^3	1001	0100	1101	0110
u^3	1101	0101	0110	1110
v^3	1001	1111	1011	0000
w^3	1110	0100	0110	1110
K^4	0100	1101	0110	0011
u^4	1010	1001	0000	1101
v^4	0110	1010	1110	1001
K^5	1101	0110	0011	1111
y	1011	1100	1101	0110

Feistel structure

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



Block Ciphers Features

- Block size: in general larger block sizes mean greater security.
- Key size: larger key size means greater security (larger key space).
- Number of rounds: multiple rounds offer increasing security.
- Encryption modes: define how messages larger than the block size are encrypted, very important for the security of the encrypted message.

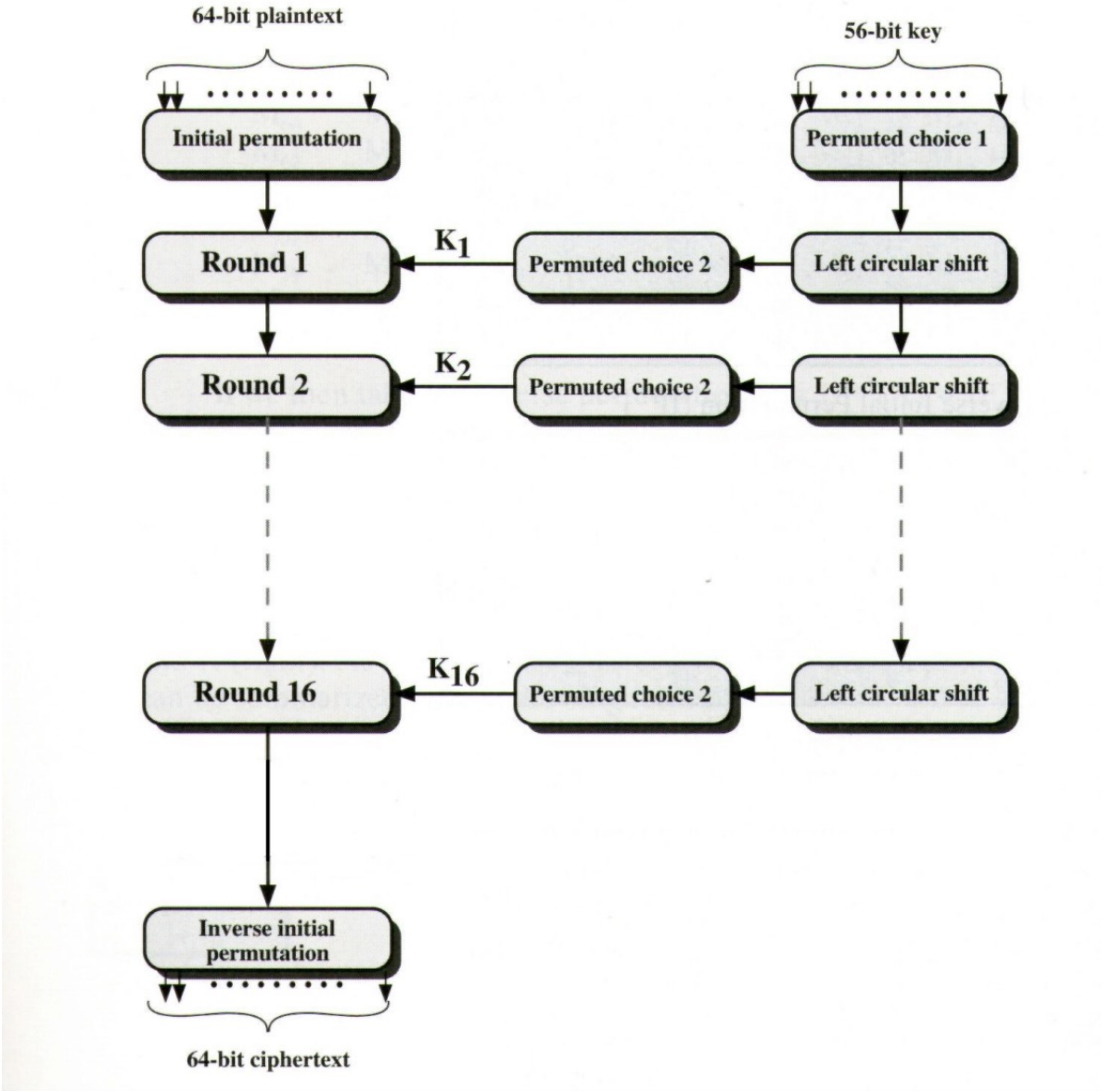
History of Data Encryption Standard (DES)

- 1967: Feistel at IBM
  - Lucifer: block size 128; key size 128 bit
- 1972: NBS asks for an encryption standard
- 1975: IBM developed DES (modification of Lucifer
  - block size 64 bits; key size 56 bits
- 1975: NSA suggests modification
- 1977: NBS adopts DES as encryption standard in (FIPS 46-1, 46-2).
- 2001: NIST adopts Rijndael as replacement to DES

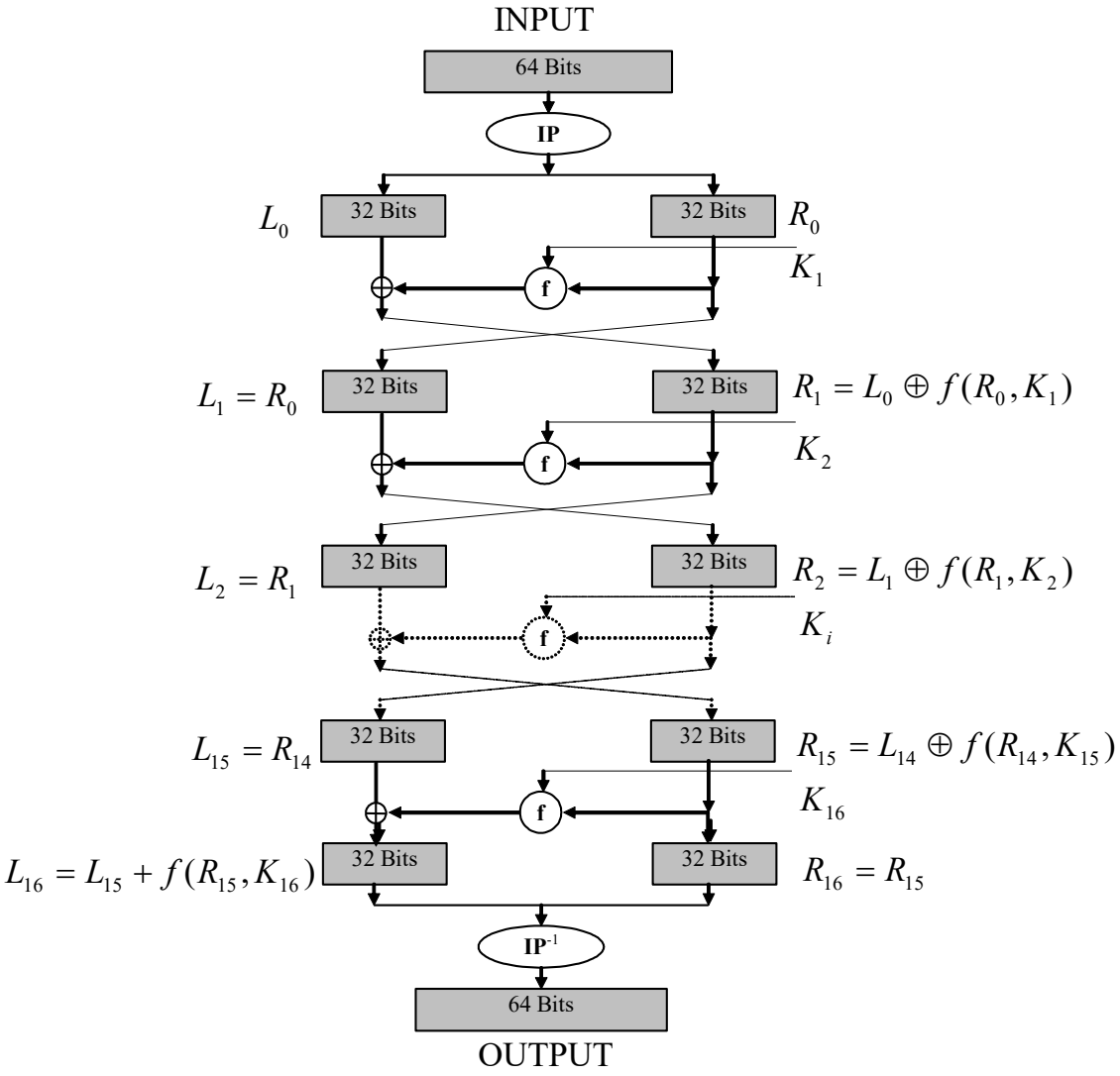
DES Features

- Block size = 64 bits
- Key size = 56 bits
- Number of rounds = 16
  - 16 sub keys, each 48 bits

DES Rounds


64-bit plaintext
56-bit key
Initial permutation
Permuted choice 1
Round 1
K1
Permuted choice 2
Left circular shift
Round 2
K2
Permuted choice 2
Left circular shift
Round 16
K16
Permuted choice 2
Left circular shift
Inverse initial permutation
64-bit ciphertext

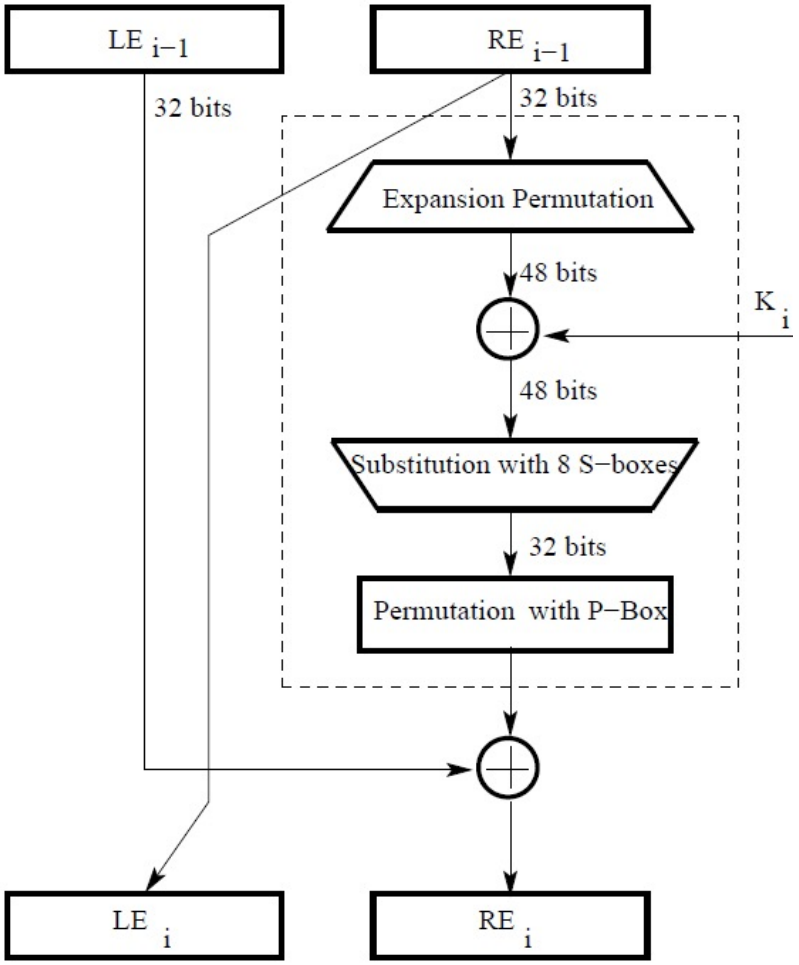
DES encryption: A closer look



Decryption uses the same algorithm as encryption, except that the subkeys $K_1, K2, \ldots K16$ are applied in reversed order

DES: f function

- RE_{i-1} is "expanded" to a bit-string of length 48: $E(A)$
- Compute $E(A) \oplus K_i$ and write the result as the concatenation of eight 6-bit strings $B = B_1B_2B_3B_4B_5B_6B_7B_8$
- Uses eight *S-boxes* $S_1, \ldots, S_8$ and compute $C_j = S_{j(B_j)}, 1 \leq j \leq 8$
- The bitstring $C = C_1C_2C_3C_4C_5C_6C_7C_8$ of length 32 is permuted according to a fixed permutation P



DES: S-boxes

- Input: $B = b_1b_2b_3b_4b_5b_6$
- Output: $C = c_1c_2c_3c_4c_5c_6$
- Two bits b_1b_6 determine the binary representation of a row r of S_j
- the four bits $b_2b_3b_4b_5$ determine the binary representation of a column c of S_j

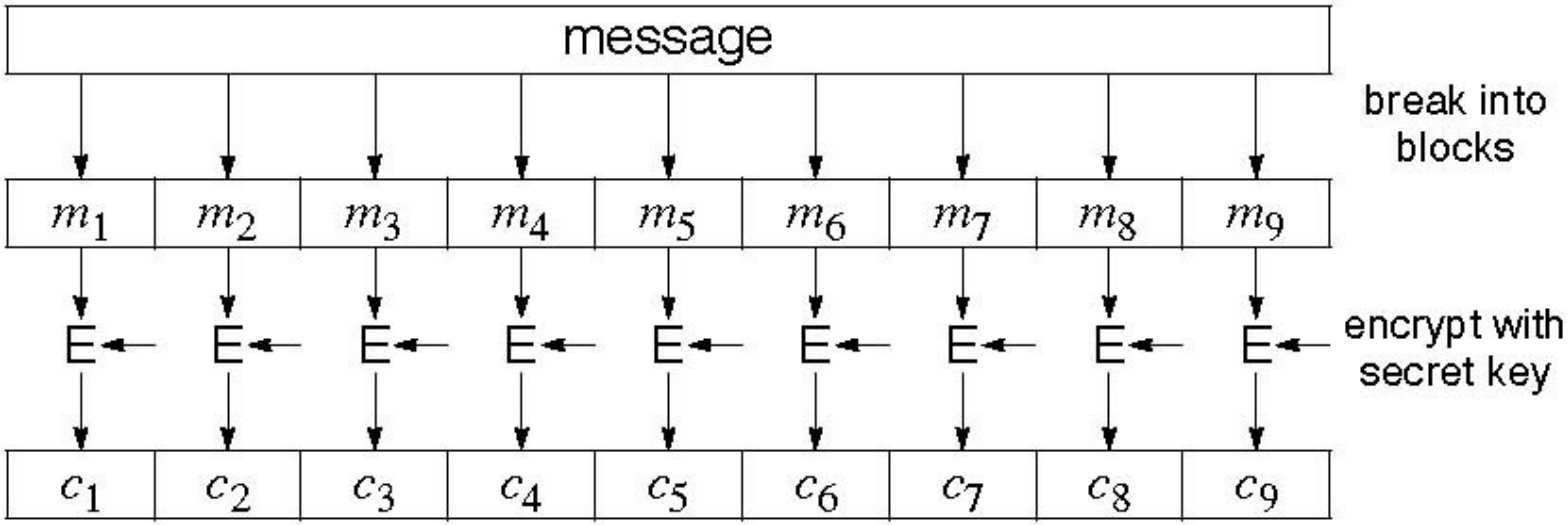
S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Modes of Operation (Encryption modes) (optional)

- Mode of operation (or encryption mode):
  - A block cipher algorithm takes on a fixed-length input, i.e. a block, and produce an output, usually a block of the same fixed-length.
  - In practice, we want to encrypt files of various length ➔ need to divide a file into block of that given fixed length ➔ then call the encryption algorithms several times
  - Operation mode: the manner and structure in which we feed the encryption algorithm (several times) with blocks of the plaintext file and concatenate the resulted blocks to produce the ciphertext file.
- The popular modes:
  - ECB, CBC, OFB, CFB
- We now overview the properties of certain modes (privacy, integrity) and potential attacks against them.

Electronic Code Book (ECB)

- Each block is independently encoded



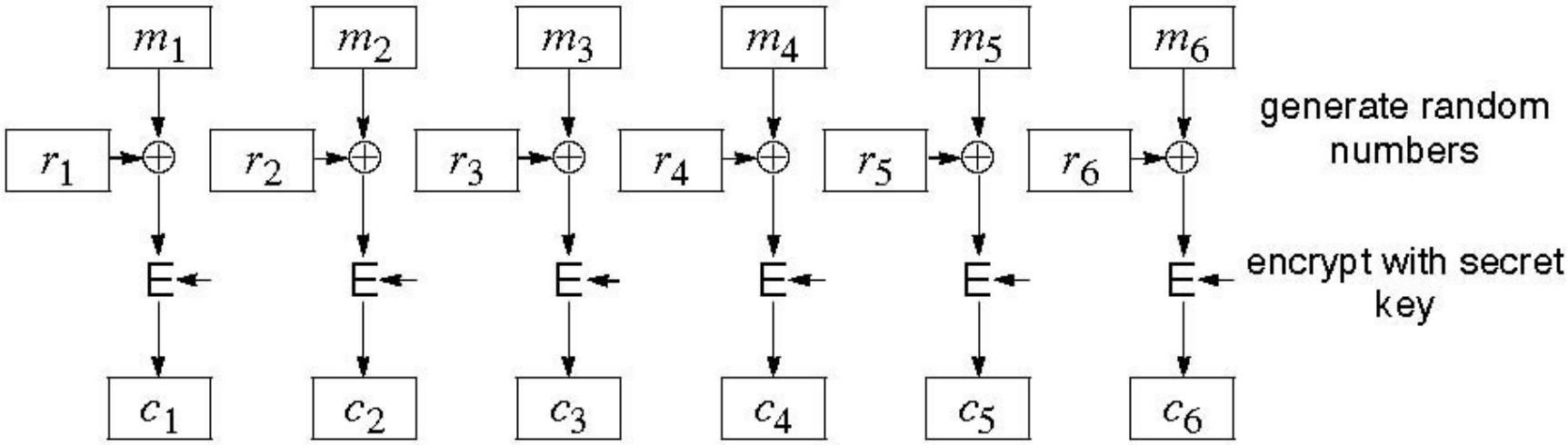
- Problem:
  - Identical Input➔ Identical Output
    - Deterministic: the same data block gets encrypted the same way, reveals patterns of data when a data block repeats.

ECB critics

- Weakness: Replay/Manipulation attack
  - Can insert encoded blocks
  - Reordering ciphertext results in reordered plaintext.
- Strength:
  - Errors in one ciphertext block do not propagate.
- Usage:
  - not recommended to encrypt more than one block of data
  - Encryption in database

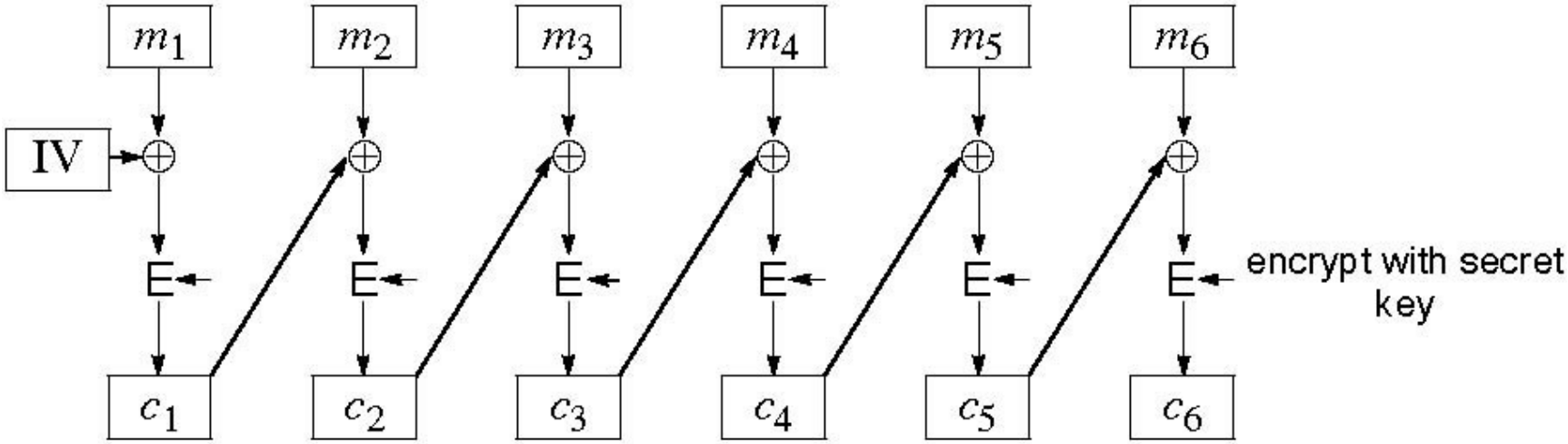
Cipher Block Chaining (CBC)

- Improving on ECB: think of adding a random number before encoding



CBC (cont.)

- The main idea:
  - Use C_i as random number block operation for i+1
  - So, need a so called Initial Value (IV)
    - If no IV, then one can guess changed blocks

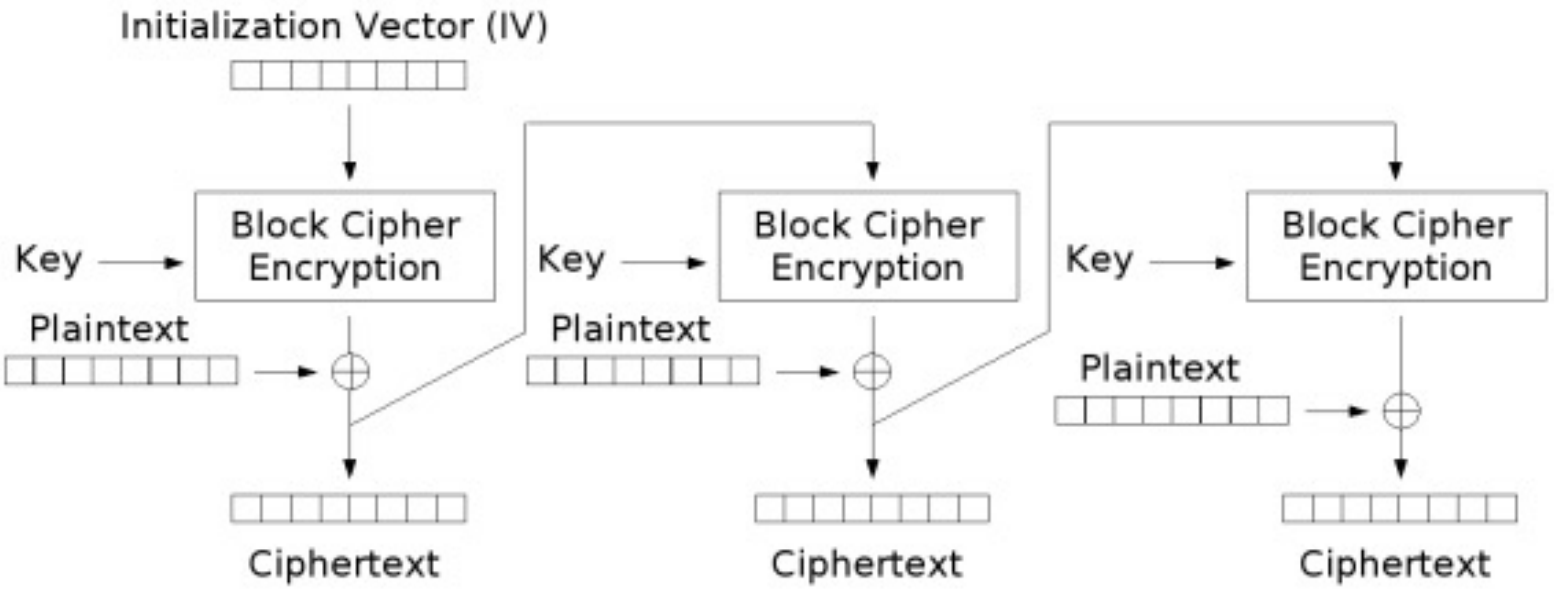


CBC critics

- Good
  - Randomized encryption: repeated text gets mapped to different encrypted data.
    - Can be proven to be “secure” assuming that the block cipher has desirable properties and that random IV’s are used
  - A ciphertext block depends on all preceding plaintext blocks
    - reorder affects decryption
- Bad
  - Errors in one block propagate to two blocks
    - one bit error in C_j affects all bits in M_j and one bit in M_j+1
  - Sequential encryption, cannot use parallel hardware
  - Observation: if $C_i=C_j$ then $E_k(M_i \oplus C_{i-1}) = E_k(M_j \oplus C_{j-1})$; thus $M_i \oplus C_{i-1} = M_j \oplus C_{j-1}$; thus $M_i \oplus M_j = C_{i-1} \oplus C_{j-1}$

Cipher Feedback (CFB)

- The message is XORed with the feedback of encrypting the previous block



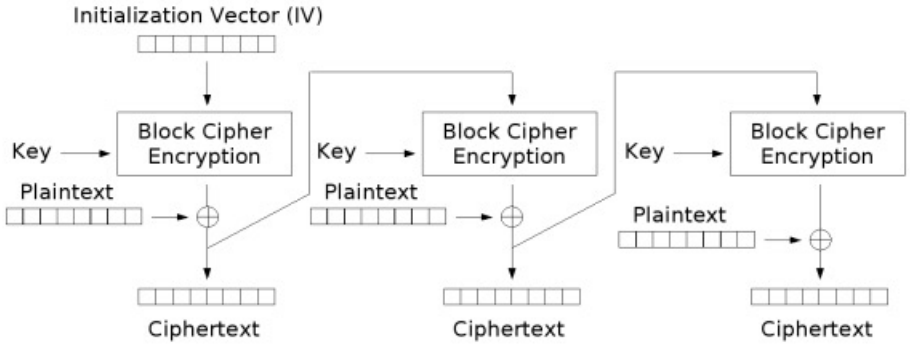
Cipher Feedback (CFB) mode encryption

CFB critics

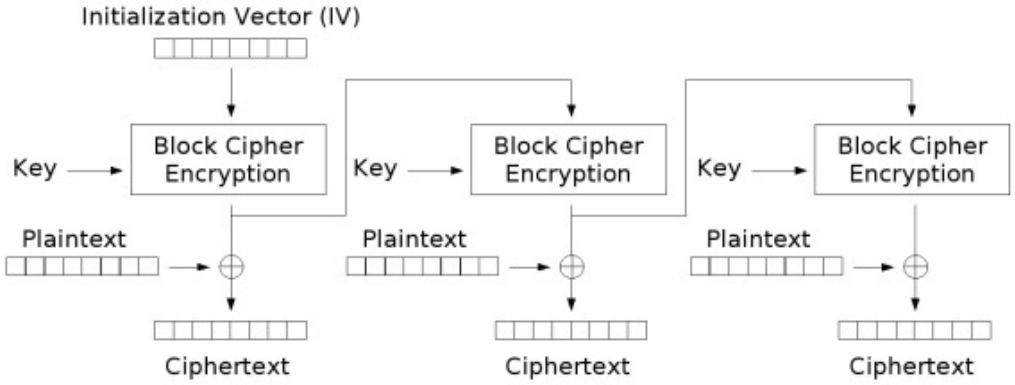
- Good
  - Randomized encryption
  - A ciphertext block depends on all preceding plaintext blocks; reorder affects decryption
- Bad
  - Errors propagate for several blocks after the error, but the mode is self-synchronizing (like CBC).
  - Sequential encryption

Output Feedback (OFB)

- IV is used to generate a stream of blocks
- Stream is used a one time pad and XOR'ed to plain text



Cipher Feedback (CFB) mode encryption



Output Feedback (OFB) mode encryption

OFB critics

- Randomized encryption
- Sequential encryption, but preprocessing possible
- Error propagation limited