Lecture 4

DES and Block Encryption modes

UTAS
Sultanate of Oman
September 2022

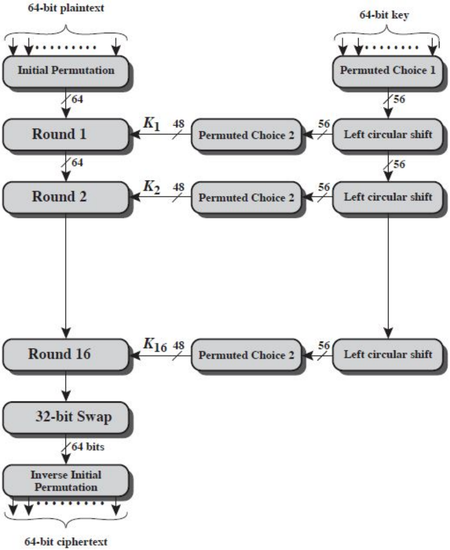
CSSY2201 : Introduction to Cryptography

Plan

DES

- Data Encryption Standard (DES) is the encryption standard recommended by NIST (National Institute of Standards and Technologies) in 1977.
- the most used encryption algorithm until 2001 (the arrival of AES by NIST too)
- The DES alg is called DEA (Data Encryption Algorithm)
- The plaintext is encrypted in 64-bit blocks using a 56-bit key size
- alg transforms a 64-bit block of plaintext to a 64-bit block of ciphertext
- The same steps, with the same key, lead to decryption

DES


64-bit plaintext
64-bit key
Initial Permutation
Permuted Choice 1
64
56
Round 1
K_1
48
Permuted Choice 2
56
Left circular shift
64
56
Round 2
K_2
48
Permuted Choice 2
56
Left circular shift
Round 16
K_{16}
48
Permuted Choice 2
56
Left circular shift
32-bit Swap
64 bits
Inverse Initial Permutation
64-bit ciphertext

Initial permutation (IP) of DES

(a) Initial Permutation (IP)

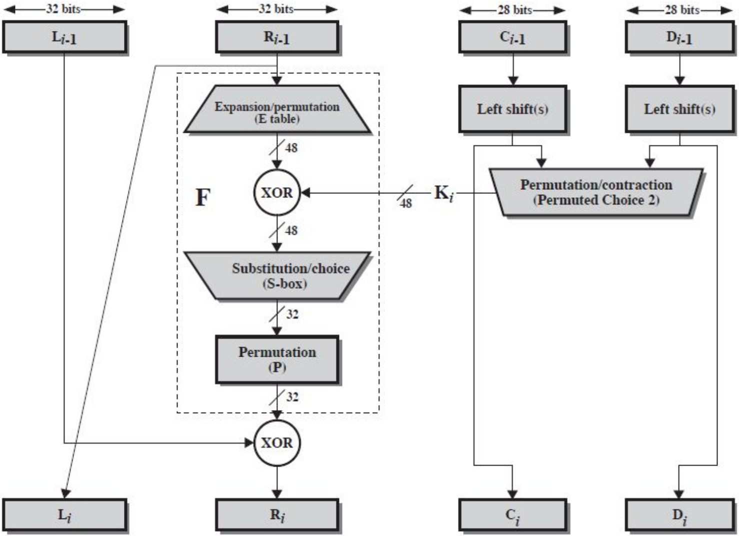
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(b) Inverse Initial Permutation (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

- ex : on the output IP :
  in pos 1 $\Longrightarrow$ input 58 ;
  in pos 2 $\Longrightarrow$ enter 50 ;
  in pos 64 $\Longrightarrow$ entry 7
- On output of IP^{-1} :
  1 is read from pos 58
  2 is read from pos 50
  64 is read from pos 7

Structure of one DES round



Structure of one DES round

- Two L and R halves of 32-bit size each
- Feistel Structure is as follows :

$$L_i = R_{i-1}$$

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

- F takes the 32-bit half R and the 48-bit round key and does the following :
  - Expansion from R to 48-bits using E permutation
  - Mix it with the round key by XOR
  - Pass it through 8 S-boxes to get the 32-bit result
  - Finally permute it using a permutation P

Permutation functions E and P

(c) Expansion Permutation (E)

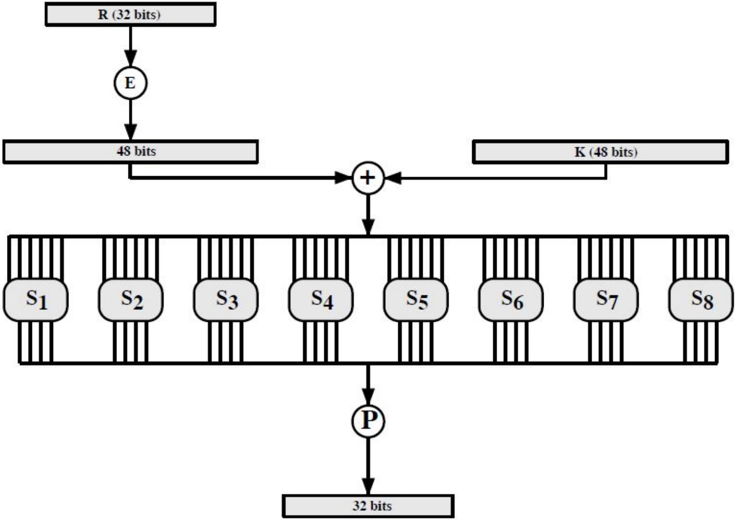
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

(d) Permutation Function (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

جامعة التقنية
والعلوم التطبيقية
University of Technology
and Applied Sciences
Salalah صلالة

Structure of one DES round : F(R,K)



The 8 elementary S-boxex

- each S-box transforms 6-bits to 4-bits
- For each entry of each S-box :
  - bits 1 and 6 (outer bits) select one row among 4.
  - bits 2-5 (inner bits) are substituted by the corresponding column in the chosen row
  - the result is 8 batches of 4-bit : that's 32-bits in all
- row selection depends on plaintext and key
- example 48 bits –> 32 bits : S(18 09 12 3d 11 17 38 39) = 5fd25e03
  - Sbox1 : 0x18 : 011000 –> line nÂ°0 and column nÂ° 12 : 5 = 0x5
  - Sbox2 : 0x09 : 001001 –> line nÂ°1 and column nÂ° 4 column : 15=0xf
  - Sbox8 : 0x39 : 111001 –> line nÂ°3 and column nÂ° 12 : 3 = 0x3

The 8 S-boxes : (1-4)

S_1	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
S_2	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
S_3	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
S_4	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

جامعة التقنية والعلوم التطبيقية
University of Technology and Applied Sciences
Salalah صلالة

The 8 S-boxes : (5-8)

S_5	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
S_6	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
S_7	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
S_8	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

جامعة التقنية والعلوم التطبيقية
University of Technology and Applied Sciences
Salalah صلالة

DES Key schedule

- Key schedule : preparation of round keys (of 16 rounds) from the original 56-bit key
- Initial permutation of key (PC1) that selects 56-bits (out of 64) into 2 halves of 28-bits
- 16 stages consisting of :
  - "Circular rotation left" of each half of 1 or 2 bits depending on the rotation function K
  - select 24 bits from each half and swap it by (PC2) to be the input of function F.

DES Key schedule

(b) Permuted Choice One (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

(c) Permuted Choice Two (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

(d) Schedule of Left Shifts

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

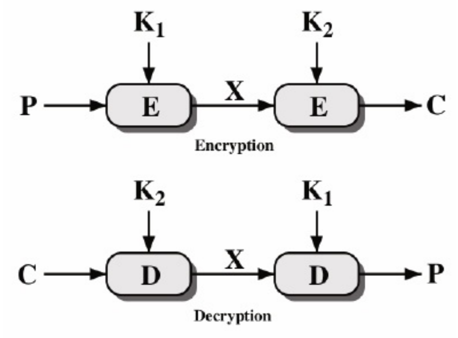
Security of DES

- keyspace size $2^{56} = 7.2 \times 10^{16}$
- a machine with 10^9 decryption/s can break it in 1.125 year
- a machine with 10^{13} decryption/s can crack it in 1 hour
- the AES-128 with the same speed, the machine remains 5.3×10^{17} years
- several attacks on DES :
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attack
- The need to find an alternative to DES becomes necessary

Multiple Encryption using DES

- DES became vulnerable to brute force attack
- Alternative : encrypt multiple times with different keys
- Options :
  - Double DES : not very efficient
  - Triple DES (3DES) with two keys : brute force 2^{112}
  - Triple DES with three keys : brute force 2^{168}

Double encryption

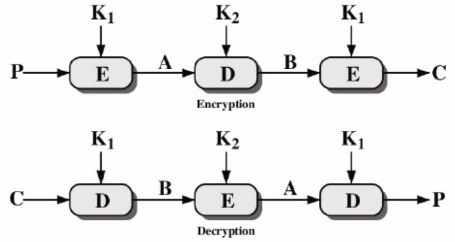


- key of size $2 \times 56 = 112$
- 2^{112} keyspace.
- so you have to try on average 2^{111} to brute force it
- smarter attack : Meet-in-the-middle attack

Meet-in-the-middle attack

- Double DES encryption : $C = E(K_2, E(K_1, P))$
- let $X = E(K_1, P) = D(K_2, C)$
- suppose the opponent knows 2 P/C pairs : (P_a, C_a) and (P_b, C_b)
  - encrypt P_a using all the possibilities 2^{56} of the key K_1 to have the possibilities of X
  - Store the possible values of X in an array with their corresponding keys K_1
  - Decrypt C_a using all possibilities 2^{56} of key K_2
  - For each decryption result, check with array values
  - If there is a match, Take the corresponding values of K_1 and K_2. And check if $C_b = E(K_2, E(K_1, P_b))$, then accept the keys.
- With two pairs of P/C, the probability of success is 1
- This attack has complexity 2×2^{56} which is much lower than brute force attack complexity 2^{112}

Triple Encryption



- 2 keys : 112 bit
- 3 keys : 168 bits
- Why E-D-E ? To be compatible with simple DES :

$$C = E(K_1, D(K_1, E(K_1, P)))$$

- 3DES has been adopted by several internet applications : PGP, S/MIME

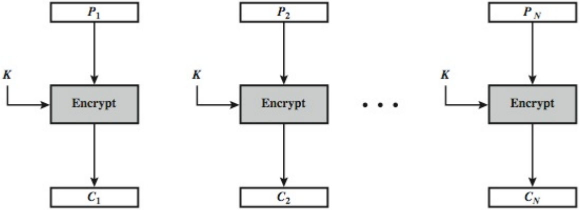
Modes Of Block Encryption

- NIST SP 800-38A define 5 modes of block encryption
  - ECB : Electronic codebook Book Mode
  - CBC : Cipher block chaining Mode
  - CFB : Cipher FeedBack Mode
  - OFB : Output FeedBack Mode
  - CTR : Counter Mode
- There are those oriented block cipher and those to be used in a stream cipher context
- This is to cover a wide variety of real-life scenario applications
- theses modes can be applied with any bloc encryption algorithm

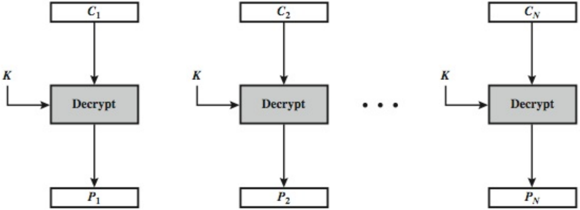
ECB

- The plaintext is divided to blocks
- Each block is substituted by an encryption mechanism like a dictionary or "code book"
- Each block is encrypted independently from the others blocs : $C_i = E_K(P_i)$
- Application : secured transmission of short messages

ECB



(a) Encryption

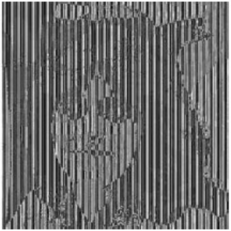


(b) Decryption

Advantages and limitations of ECB

- Repeats in plaintext are also shown in ciphertext (little confusion)
- Not effective for images : too much redundancy, too many repetitions so image can remain visible after encryption
- The weakness is in the independence in the encryption of the different blocks
- Main use is very short plaintext encryption

Original image



Encrypted image (AES) in ECB mode

CBC

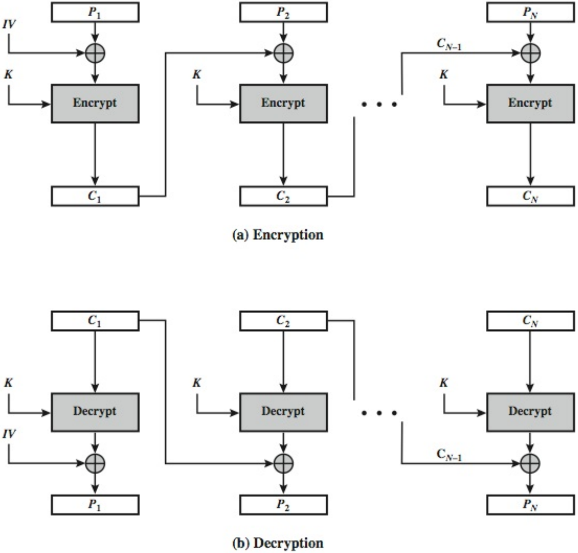
- The plaintext is divided into blocks
- these blocks will be linked during encryption
- each ciphertext block is linked with the corresponding plaintext block and previous ciphertext blocks
- uses an initialization vector to begin encryption :

$$C_i = E_K(P_i \; XOR \; C_{i-1})$$

$$C_{-1} = IV$$

- Application : Bulk data encryption (high redundancy) ; Authentication (CMAC)

CBC


P_1
P_2
P_N
IV
K
Encrypt
C_{N-1}
C_1
C_2
C_N
(a) Encryption
Decrypt
(b) Decryption

Advantages and limitations of CBC

- each block in the ciphertext depends on all the blocks before it
- any change affects all ciphertext blocks that follow it
- CBC needs a **IV** for initialization :
  - the IV must be known to sender and receiver
  - If transmitted in the clear, an adversary can change the bits of the first block and change IV to compensate for this change.
  - So IV must be either fixed
  - is sent encrypted in ECB mode before processing the plaintext

Stream-oriented block encryption modes

- Block modes encrypts the whole block
- in some applications, we may need to operate on smaller sizes
- application in media stream encryption (real time)
- convert block algs to stream algs
  - CFB
  - OFB
  - CTR
- the idea is to use block algs as a pseudo-random sequence generator

CFB

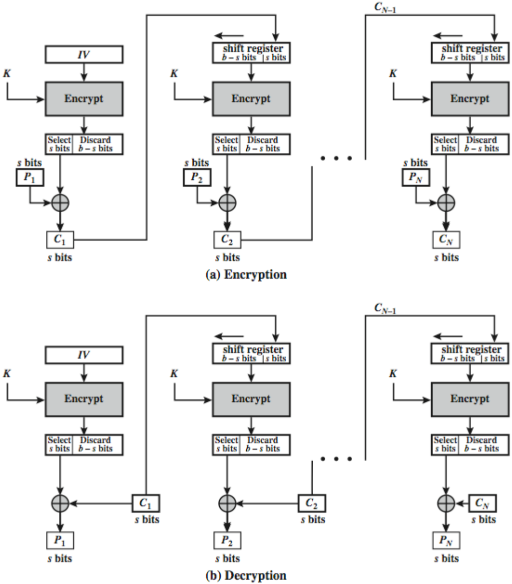
- The message is treated as a bit stream
- message is added to block alg output
- the result is feedback to the next stage
- the standard allows several block sizes 1, 8, 64, 128, etc to be feedback noted CFB-1, CFB-8, CFB-64, CFB-128
- encryption is as follows

$$C_i = P_i \, XOR \, E_K(C_{i-1})$$

$$C_{-1} = IV$$

- Applications : stream encryption (real time), Authentication

CFB


IV
K
Encrypt
Select s bits
Discard b – s bits
s bits
P_1
C_1
s bits
shift register
b – s bits
s bits
P_2
C_2
C_{N-1}
P_N
C_N
(a) Encryption
(b) Decryption

Advantages and limitations of CFB

- CFB is appropriate if the data arrives in bits or bytes
- appropriate for streaming mode
- Note that in encryption and decryption, both operate with the cipher block E_K
- the error (if any) can propagate in several blocks after the erroneous block

OFB

- The message is treated as a bit stream
- encryption output is added to the message
- The Output is feedback to the input of the next stage
- the feedback is independent of the message (plaintext)
- it can be calculated before

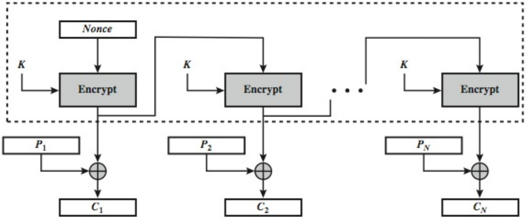
$$O_i = E_K(O_{i-1})$$

$$C_i = P_i \, XOR \, O_i$$

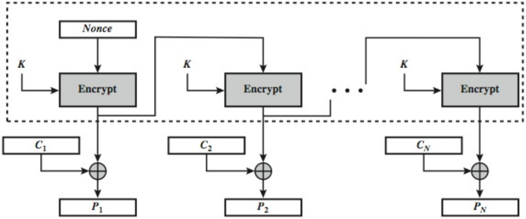
$$O_{-1} = IV$$

- Usage : Stream encryption in a noisy channel

OFB



(a) Encryption



(b) Decryption

Advantages and Limitations of OFB

- OFB needs an IV which must be unique for each use
- if the IV is reused, the adversary can find the outputs
- Errors do not propagate
- sender and receiver must be synchronized

CTR

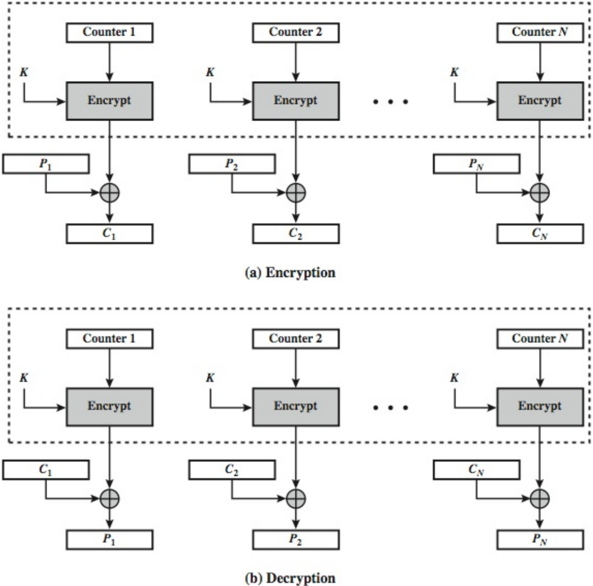
- a new mode similar to OFB but encrypts a counter instead of the output
- must have a different key and counter value for each post

$$O_i = E_K(i)$$

$$C_i = P_i \, XOR \, O_i$$

- Usage : Encryption in high-speed networks

CTR


Counter 1
Counter 2
Counter N
K
Encrypt
. . .
P1
P2
PN
C1
C2
CN
(a) Encryption
Counter 1
Counter 2
Counter N
K
Encrypt
. . .
C1
C2
CN
P1
P2
PN
(b) Decryption