CS408
Cryptography & Internet Security

Lectures 6, 7, 8:

DES,

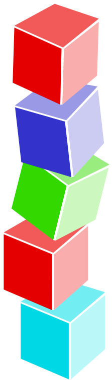
DES Security,

Modes of operation

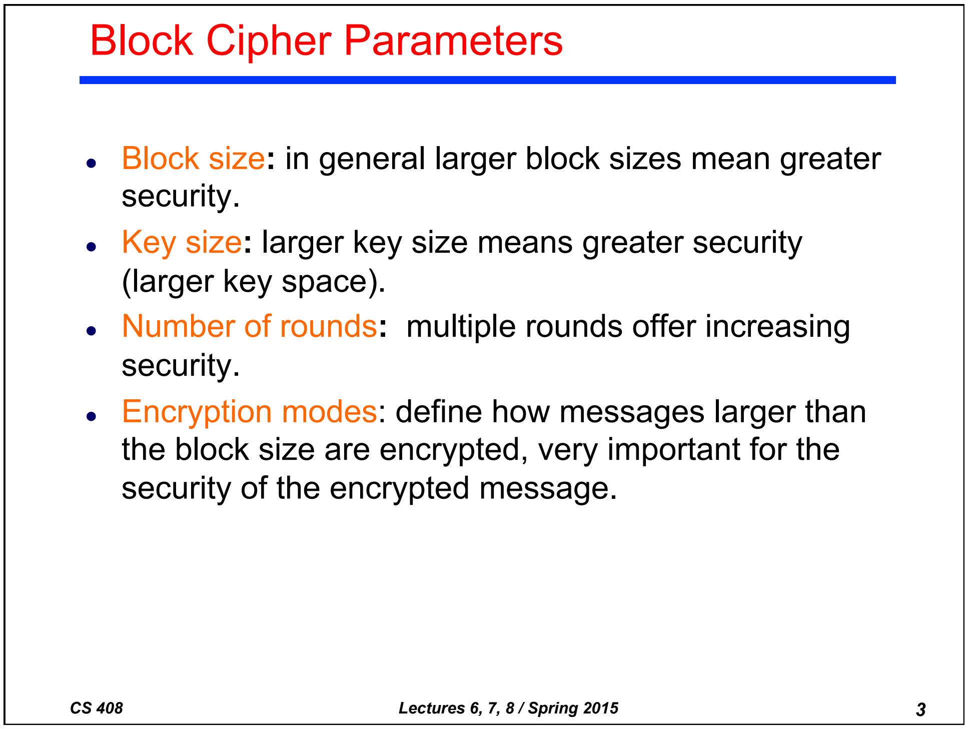
Reza Curtmola
Department of Computer Science
NJIT

Block Cipher

- Maps n-bit plaintext blocks to n-bit ciphertext blocks (n: block length)
  - $\mathcal{P} = \mathcal{C} = \{0,1\}^n$

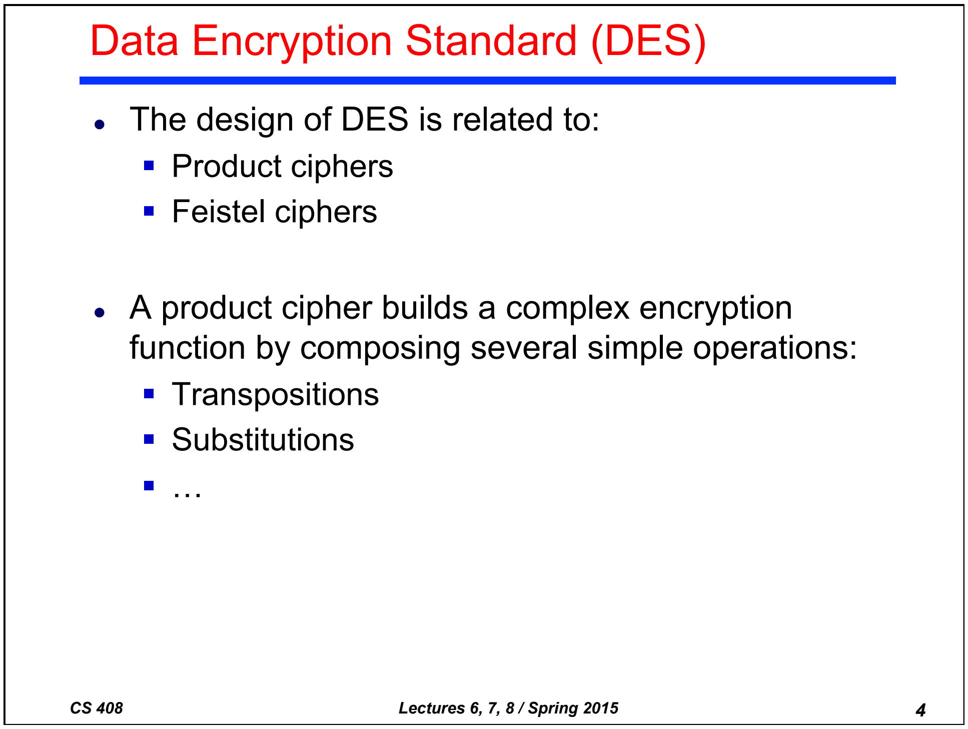


- $y = E_k(x)$ denotes the encryption of plaintext x under key k to obtain ciphertext y
- $x = D_k(y)$ denotes the decryption of ciphertext y under key k to obtain the plaintext x
- The encryption function E is a bijection
  - it is invertible
  - there is a one-to-one mapping between plaintexts and ciphertexts - this allows for unique decryption



Block Cipher Parameters

- 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.



Data Encryption Standard (DES)

- The design of DES is related to:
  - Product ciphers
  - Feistel ciphers

- A product cipher builds a complex encryption function by composing several simple operations:
  - Transpositions
  - Substitutions
  - …

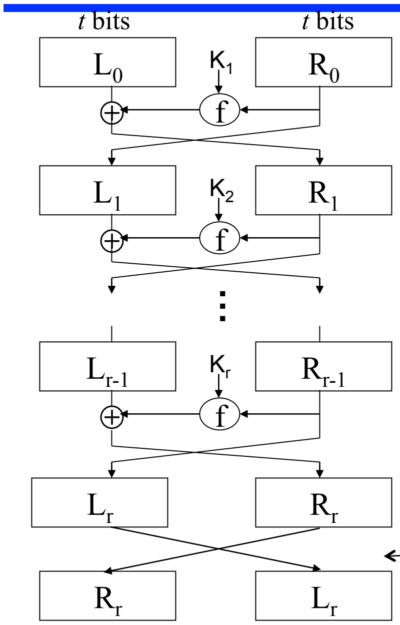
Feistel cipher (Feistel network)

- Iterated block cipher is a block cipher involving the sequential repetition of an internal function called a round function
- Feistel cipher (Feistel network)
  - Iterated block cipher
  - Maps a 2t-bit plaintext (L_0, R_0), where L_0, R_0 are t-bit blocks, into a ciphertext (L_r, R_r), through an r-round process
  - In each round i, for $1 \le i \le r$, (L_{i-1}, R_{i-1}) is mapped into (L_i, R_i) using the round key K_i as follows:

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

    where each round key K_i is derived from the cipher key K
  - Used in DES, IDEA, RC5, and many other block ciphers.
  - Not used in AES

Feistel Cipher



Encryption:

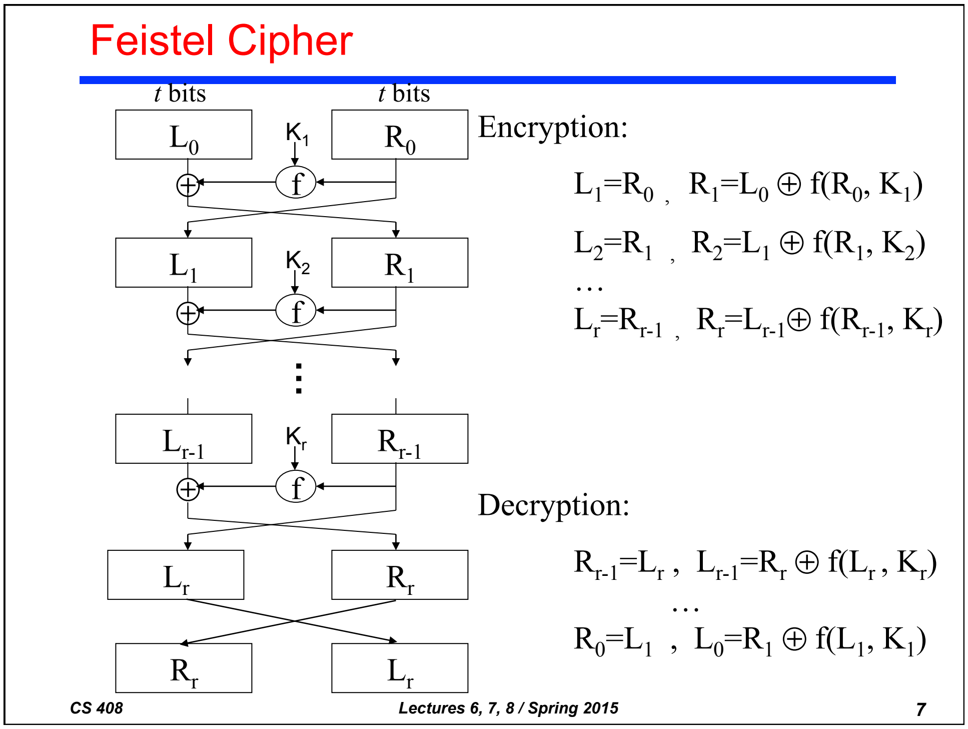
$L_1=R_0$, $R_1=L_0 \oplus f(R_0, K_1)$

$L_2=R_1$, $R_2=L_1 \oplus f(R_1, K_2)$

…

$L_r=R_{r-1}$, $R_r=L_{r-1} \oplus f(R_{r-1}, K_r)$

The two halves are flipped after the last round



Feistel Cipher

Encryption:

$L_1=R_0$, $R_1=L_0 \oplus f(R_0, K_1)$

$L_2=R_1$, $R_2=L_1 \oplus f(R_1, K_2)$

…

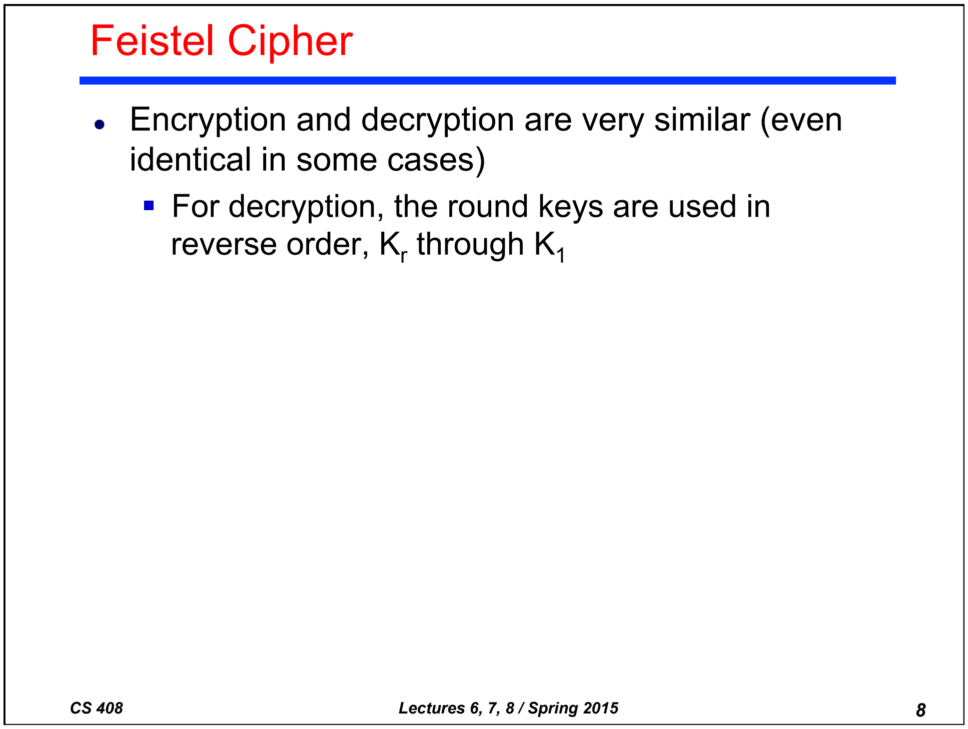
$L_r=R_{r-1}$, $R_r=L_{r-1} \oplus f(R_{r-1}, K_r)$

Decryption:

$R_{r-1}=L_r$, $L_{r-1}=R_r \oplus f(L_r , K_r)$

…

$R_0=L_1$, $L_0=R_1 \oplus f(L_1, K_1)$



Feistel Cipher

- Encryption and decryption are very similar (even identical in some cases)
  - For decryption, the round keys are used in reverse order, K_r through K_1

A Word About NIST and Standards

- NIST = National Institute of Standards and Technology
- “Founded in 1901, NIST (former NBS - National Bureau of Standards) is a non-regulatory federal agency within the U.S. Commerce Department’s Technology Administration. NIST's mission is to develop and promote measurement, standards, and technology to enhance productivity, facilitate trade, and improve the quality of life.”
- Cryptographic Standards & Applications
- Federal Information Processing Standards (FIPS): define security standards

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: DES was developed at IBM and first published in the Federal Register of March 17, 1975
  - block size 64 bits; key size 56 bits
- 1975: NSA suggests modification

(http://en.wikipedia.org/wiki/Data_Encryption_Standard#NSA.27s_involvement_in_the_design)

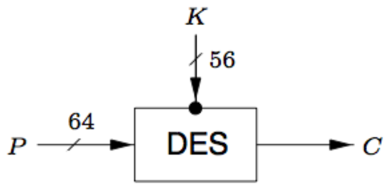
- 1977: NBS adopts DES as encryption standard in (FIPS 46-1, 46-2).
- 2001: NIST adopts Rijndael as replacement to DES.

DES Features

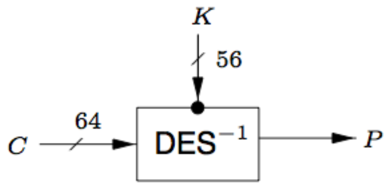
- **Features:**
  - **It is a Feistel cipher**
  - **Block size = 64 bits**
  - **Key size = 56 bits**
  - **Number of rounds = 16**
  - **16 round keys, each has 48 bits**

DES Encryption and Decryption

- Encryption



- Decryption



DES Details

$(L_0, R_o) = IP(m_1 \ldots m_{64})$

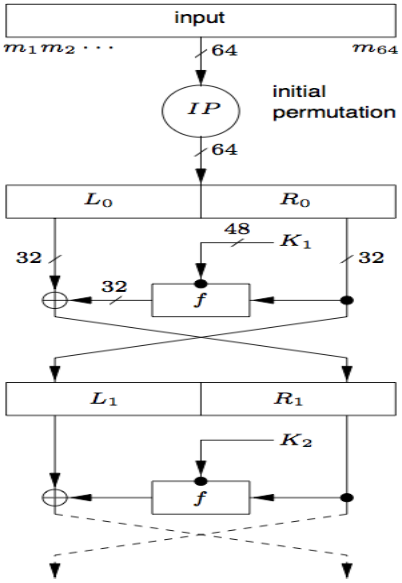
$L_i = R_{i-1}$

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

where:

IP = initial permutation

f = round function



DES Details (2)

For 16 rounds:

$L_i = R_{i-1}$

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

$b_1 \ldots b_{64} = (R_{16}, L_{16})$

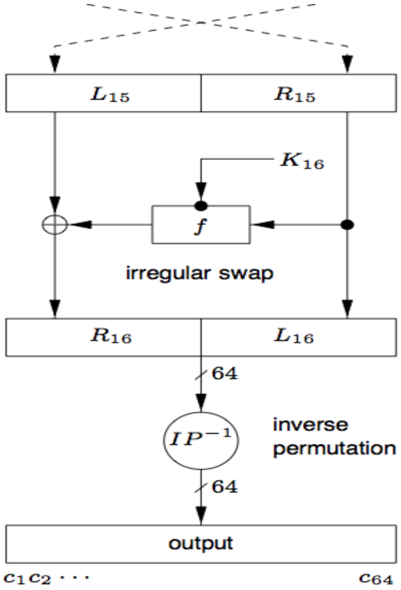
(swap final blocks L_{16}, R_{16})

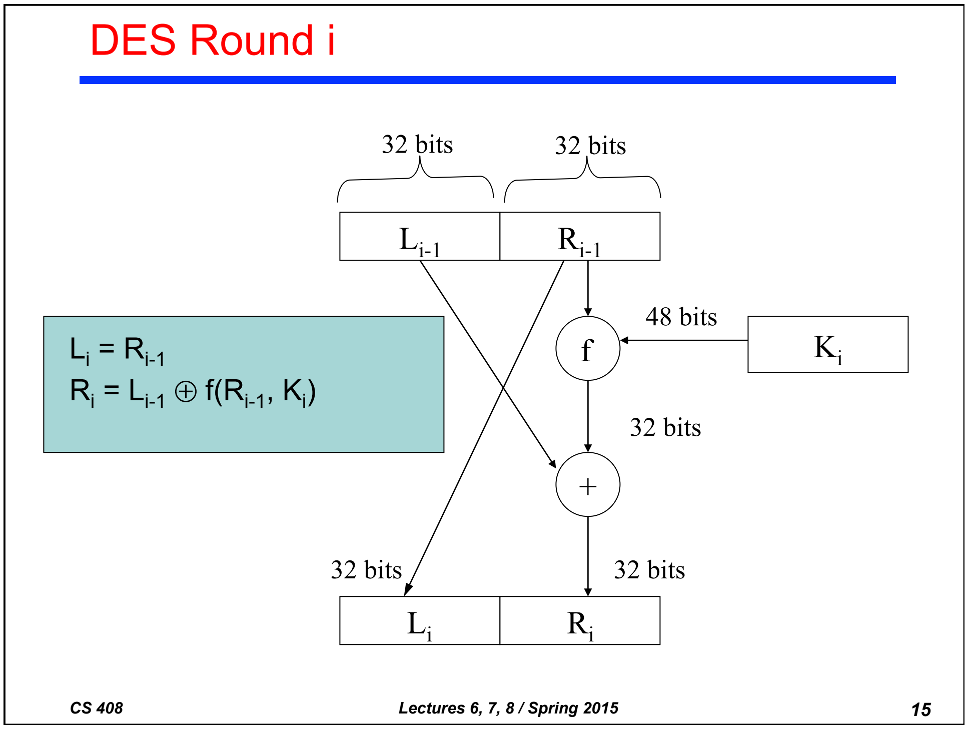
$C = (c_1 \ldots c_{64}) =$

$= IP^{-1}(b_1 \ldots b_{64})$

where:

IP^{-1} = inverse of initial permutation

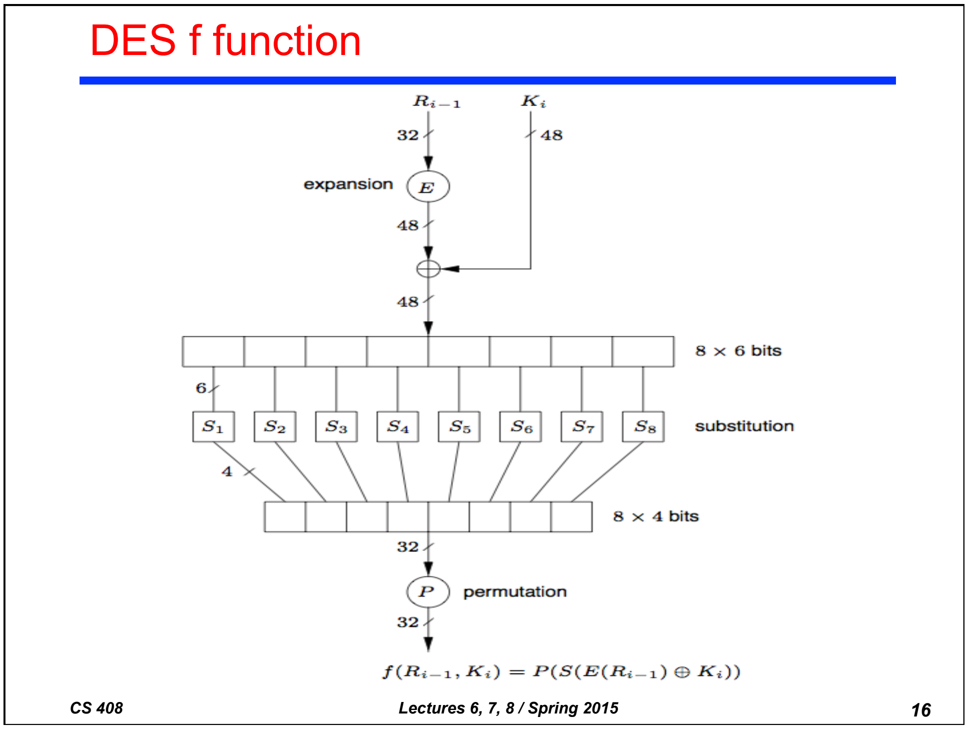




DES Round i

$L_i = R_{i-1}$

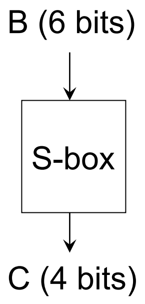
$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$



DES f function

$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$

S-boxes



- S-boxes are the only non-linear elements in DES design
- $B = b_1b_2b_3b_4b_5b_6$
- $C = c_1c_2c_3c_4$
- Each S-box S is a 4 x 16 matrix, with values from 0 to 15 (i.e., each entry $S[i,j] \in [0, 15]$, and has 4 bits)
- In each S, rows are numbered from 0 to 3, and columns are numbered from 0 to 15
- C is obtained as the entry in row i and column j of matrix S
  - $i = 2b_1 + b_6$
  - j = integer value of binary bitstring $b_2b_3b_4b_5$

There are 8 S-boxes, S_1 through S_8

S-boxes: Example

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[i, j] <16, can be represented with 4 bits

B =101111

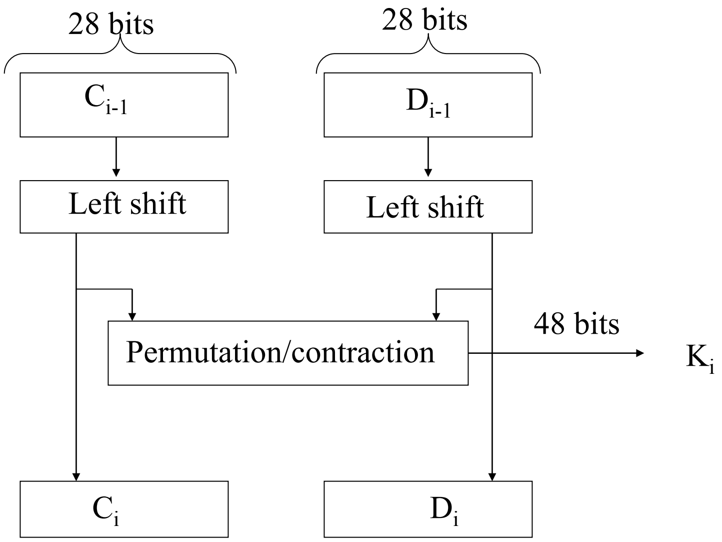
$b_1b_6 = 11$ => row 3

$b_2b_3b_4b_5 = 0111$ => column 7

DES Key Generation Schedule

- From the cipher key K, 16 round keys are generated: K_1, K_2, …, K_{16} (also known as subkeys)
- The cipher secret key K is specified as a 64-bit key, but the effective size is 56 bits, because 8 bits are parity bits (bits 8, 16, …, 64)
    - For every 7 bits of the 56 bits, an extra bit is added as the parity bit (least significant bit) such that the number of “1” bits in the byte is an odd number
- Each round key K_i contains 48 bits obtained from K

DES Key Generation Schedule (2)



28 bits

28 bits

C_{i-1}

D_{i-1}

Left shift

Left shift

Permutation/contraction

48 bits

K_i

C_i

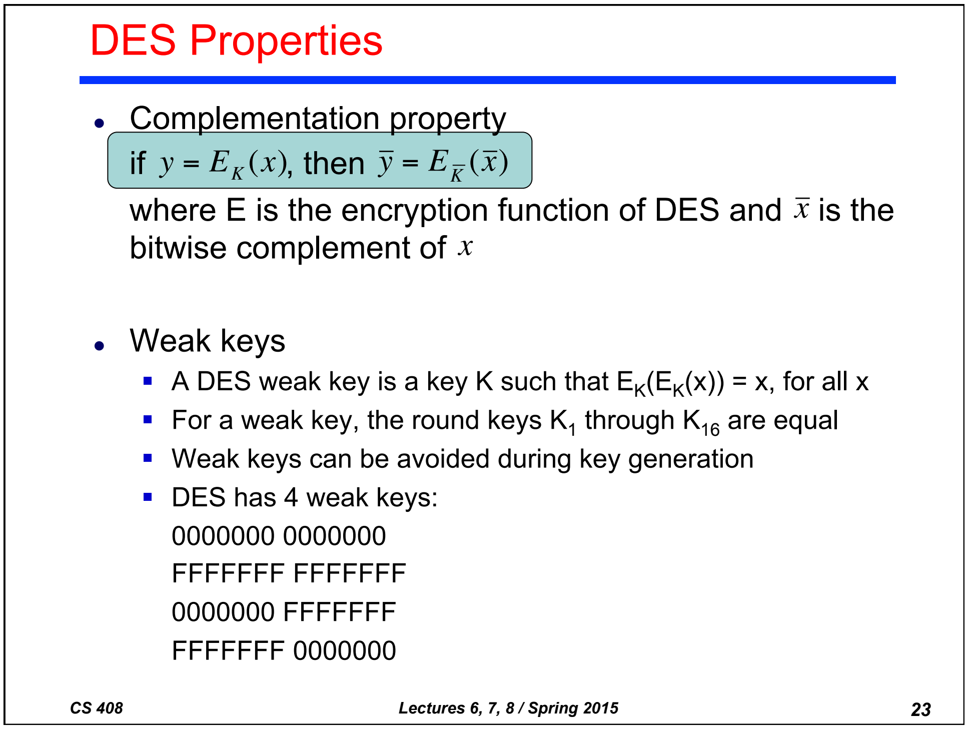
D_i

DES Decryption

- Decryption uses the same algorithm as encryption, except that the round keys $K_1, K_2, \ldots K_{16}$ are applied in reverse order: $K_{16}, K_{15}, \ldots, K_2, K_1$
- **WHY DOES THE DECRYPTION WORK?**

Desirable Objectives of Block Ciphers

- Each bit of **C** depends on all bits of **K** and **P**
- No obvious statistical relationship between **C** and **P**
- Altering any bit of **P** or **K** should alter each bit of **C** with probability 1/2
- Altering a bit of **C** should result in an unpredictable change to the recovered plaintext **P**
  - Resistance to malleability attacks
- Empirically, DES satisfies all these objectives



DES Properties

- Complementation property

  if $y = E_K(x)$, then $\bar{y} = E_{\bar{K}}(\bar{x})$

  where E is the encryption function of DES and $\bar{x}$ is the bitwise complement of x

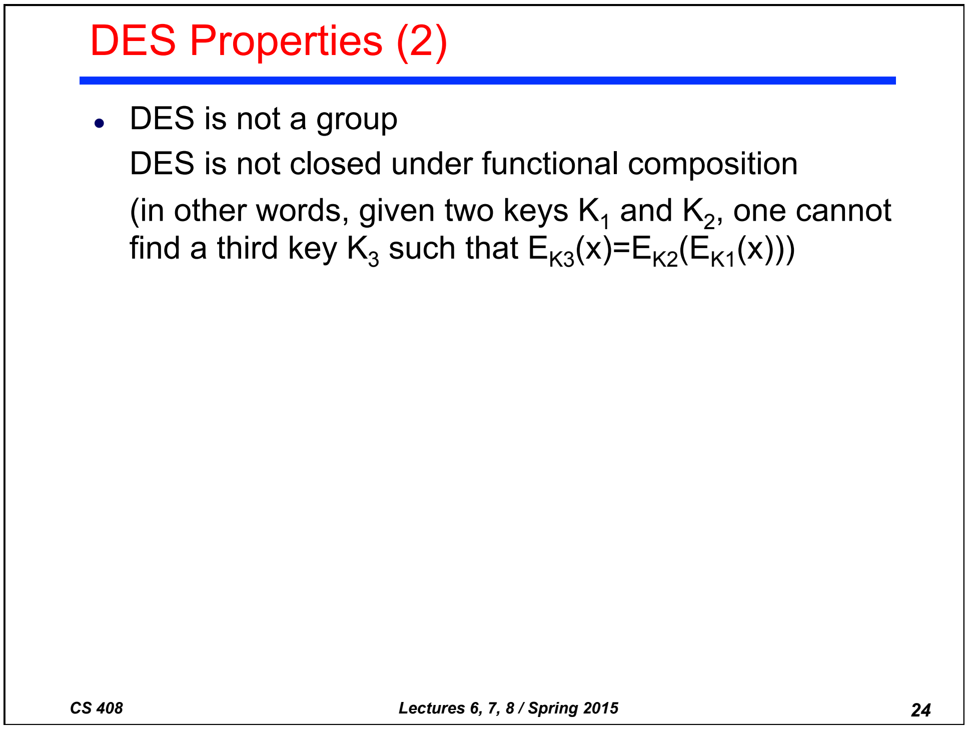
- Weak keys
  - A DES weak key is a key K such that $E_K(E_K(x)) = x$, for all x
  - For a weak key, the round keys K_1 through K_{16} are equal
  - Weak keys can be avoided during key generation
  - DES has 4 weak keys:

    0000000 0000000

    FFFFFFF FFFFFFF

    0000000 FFFFFFF

    FFFFFFF 0000000

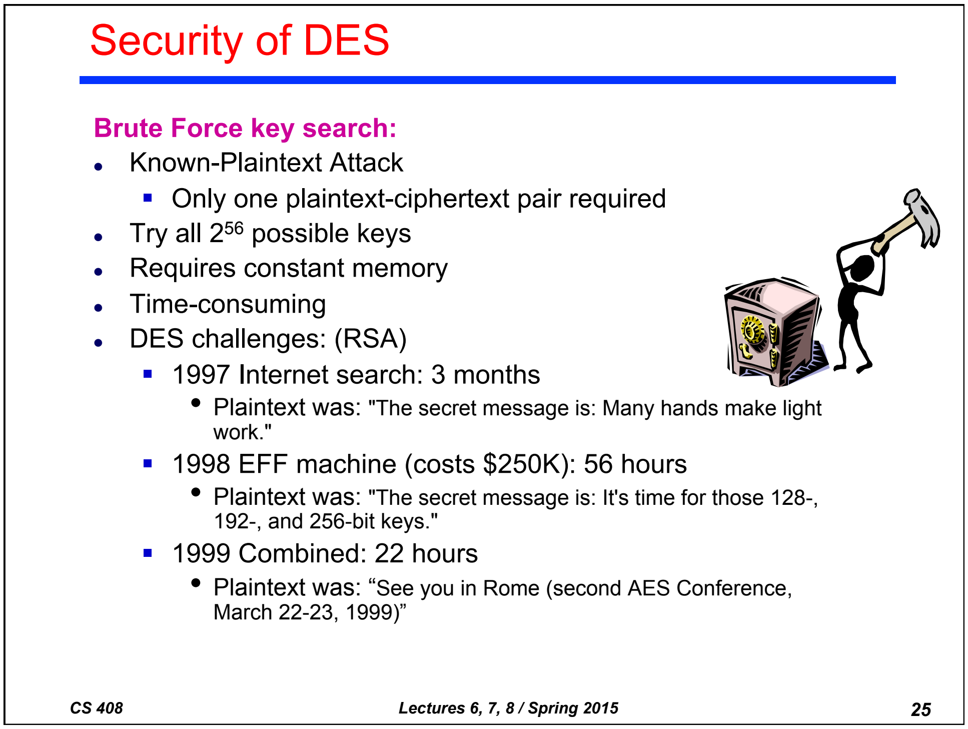


DES Properties (2)

- DES is not a group

  DES is not closed under functional composition

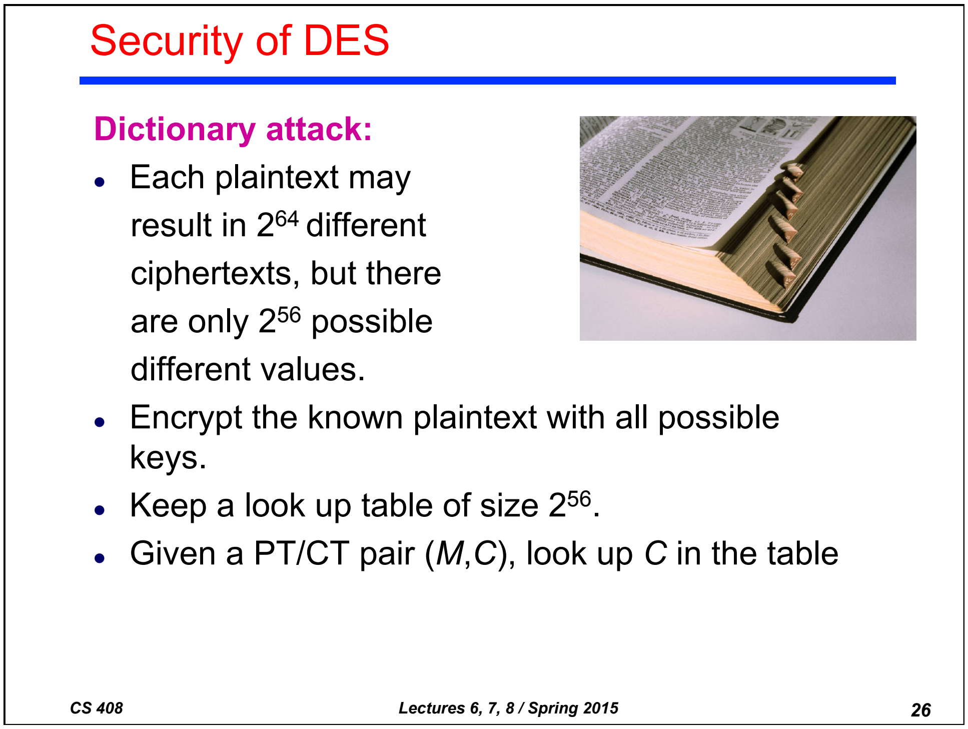
  (in other words, given two keys K_1 and K_2, one cannot find a third key K_3 such that $E_{K3}(x)=E_{K2}(E_{K1}(x))$)



Security of DES

Brute Force key search:

- Known-Plaintext Attack
  - Only one plaintext-ciphertext pair required
- Try all 2^{56} possible keys
- Requires constant memory
- Time-consuming
- DES challenges: (RSA)
  - 1997 Internet search: 3 months
    - Plaintext was: "The secret message is: Many hands make light work."
  - 1998 EFF machine (costs $250K): 56 hours
    - Plaintext was: "The secret message is: It's time for those 128-, 192-, and 256-bit keys."
  - 1999 Combined: 22 hours
    - Plaintext was: “See you in Rome (second AES Conference, March 22-23, 1999)”



Security of DES

Dictionary attack:

- Each plaintext may result in 2^{64} different ciphertexts, but there are only 2^{56} possible different values.
- Encrypt the known plaintext with all possible keys.
- Keep a look up table of size 2^{56}.
- Given a PT/CT pair (M,C), look up C in the table

Double DES

- DES uses a 56-bit key, this raised concerns about brute force attacks
- One proposed solution: double DES
- Apply DES twice using two keys, K_1 and K_2

$$C = E_{K_2}(E_{K_1}(P))$$
$$P = D_{K_1}(D_{K_2}(C))$$

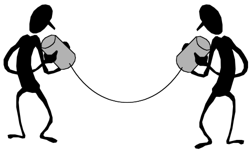
- This leads to a 2x56=112 bit key, so it seems to be twice more secure than DES.

Is it?

Meet-in-the-Middle Attack

- Goal: given one pair (P, C), find keys K_1 and K_2.
- Based on the observation:

$$\mathbf{C = E_{K_2}(\,E_{K_1}(\,P\,)\,)}$$
$$\mathbf{D_{K_2}(\,C\,) = E_{K_1}(\,P\,)}$$



- The attack has higher chance of succeeding if a second pair (P’ , C’) is available to the cryptanalyst

Meet-in-the-Middle Attack (cont.)

$C = E_{K_2}(E_{K_1}(P))$
$E_{K_1}(P) = D_{K_2}(C)$

Attack assumes the attacker knows two pairs (P,C) and (P’,C’):

- Encrypt P with all 2^{56} possible keys K_1
- Store all pairs (K_1, $E_{K_1}(P)$) in a table, sorted by $E_{K_1}(P)$
- Decrypt C using all 2^{56} possible keys K_2
- For each decrypted result, check sorted table to see if there is a match $D_{K_2}(C) = E_{K_1}(P)$
- If yes, try another pair (P’ , C’) for the keys (K_1, K_2)
- If a match is found on the new pair, accept the keys K_1 and K_2

Why Two Pairs (P, C)?

- DES encrypts 64-bit blocks, so for a given plaintext P, there are 2^{64} potential ciphertexts C.
- Key space: two 56-bit key, so there are 2^{112} potential double keys that can map P to C.
- Given a pair (P, C), the number of double keys (K_1, K_2) that produce $C = E_{K_2}(E_{K_1}(P))$ is at most $2^{112}/2^{64} = 2^{48}$
- Therefore, for a pair (P, C), 2^{48} false alarms are expected.

Why Two Pairs (P, C)? (cont.)

- With one more pair (P' , C'), extra 64-bit of known text, the alarm rate is $2^{48}/ 2^{64} = 1/2^{16} = 2^{-16}$
- If meet-in-the-middle is performed on two pairs (P, C) and (P' , C'), the correct keys K_1 and K_2 can be determined with probability $1 - 1/2^{16.}$
- Known plaintext attack against double DES succeeds in 2^{56} as opposed to 2^{55} for DES (on average).
- **The 112-bit key provides a security level similar to the 56-bit key.**

Triple DES (3DES)

Encrypt: $C = E_{K_3}(D_{K_2}(E_{K_1}(P)))$

Decrypt: $P = D_{K_1}(E_{K_2}(D_{K_3}(C)))$

- Three keying options:
  1. $K_1 \neq K_2 \neq K_3$
     (key space is 56 x 3 = 168 bits)
  2. $K_1 = K_3 \neq K_2$
     (key space is 56 x 2 = 112 bits)
  3. $K_1 = K_2 = K_3$
     (equivalent to simple DES, key space = 56 bits provides backward compatibility with DES)

- No known practical attack against keying option 1.
- Many protocols/applications use 3DES (example PGP)

Differential Cryptanalysis

- Main idea:
  - This is a chosen plaintext attack, assumes than an attacker knows (plaintext, ciphertext) pairs
  - Difference $\Delta_P = P_1 \oplus P_2$, $\Delta_C = C_1 \oplus C_2$
  - Distribution of Δ_C's given Δ_P may reveal information about the key (certain key bits)
  - After finding several bits, use brute-force for the rest of the bits to find the key.

Differential Cryptanalysis of DES

- Surprisingly … DES was resistant to differential cryptanalysis.
- At the time DES was designed, the authors knew about differential cryptanalysis. S-boxes were designed to resist differential cryptanalysis.
- Against 8-round DES, attack requires 2^{38} known plaintext-ciphertext pairs.
- Against 16-round DES, attack requires 2^{47} chosen plaintexts.
- Differential cryptanalysis not effective against DES in practice.

Linear Cryptanalysis of DES

- Another attack described in 1993 by M. Matsui
- Instead of looking for isolated points at which a block cipher behaves like something simpler, it involves trying to create a simpler approximation to the block cipher as a whole.
- It is an attack that can be applied to an iterated cipher.

Linear Cryptanalysis of DES

- M. Matsui showed (1993/1994) that DES can be broken:
  - 8 rounds: 2^{21} known plaintext
  - 16 rounds: 2^{43} known plaintext

  The attack has no practical implication, requires too many pairs.

- Exhaustive search remains the most effective attack against DES

DES Strength Against Various Attacks

Attack Method	Known Plaintext	Chosen Plaintext	Storage complexity	Processing complexity
Exhaustive precomputation	-	1	2^{56}	1
Exhaustive search	1	-	negligible	2^{55}
Linear cryptanalysis	2^{43} 2^{38}	- -	For texts	2^{43} 2^{50}
Differential cryptanalysis	- 2^{55}	2^{47} -	For texts	2^{47} 2^{55}

The weakest point of DES remains the size of the key (56 bits)!

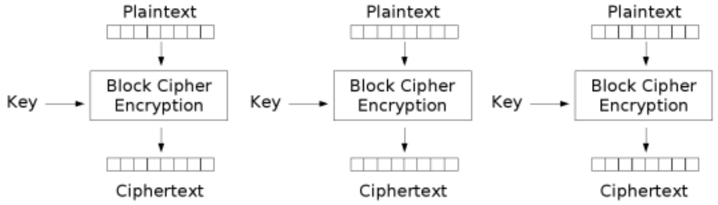
Block Cipher Modes of Operation

- What happens if the plaintext is larger than the block size?
- A t-bit plaintext message is split into n-bit blocks (n is the size of each block)
  - The message is encrypted block by block
- How is the encryption of each plaintext block combined to produce the final ciphertext message?
- Using block cipher modes of operation!
  - Electronic Code Book (ECB)
  - Cipher-block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter Mode (CTR)

Electronic Code Book (ECB)

- t = size of plaintext; n = size of each block;
  m = t / n = number of blocks
- Each block is encrypted/decrypted separately

- Encryption: $c_i = E_K(p_i)$, for $1 \le i \le m$
- Decryption: $p_i = D_K(c_i)$, for $1 \le i \le m$



ECB mode encryption

Properties of ECB mode

☹ Identical plaintext blocks result in identical ciphertext blocks

- Is this good for security?

☹ Reordering ciphertext blocks results in reordered plaintext blocks

- We say the cipher is malleable

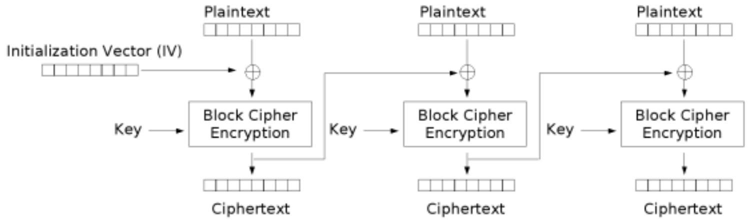
☺ Errors in one ciphertext block do not propagate to other blocks

☺ Encryption and decryption can be parallelized

- Usage: not recommended for encrypting more than one block of data

Cipher-block Chaining (CBC)

- Encryption: $c_i = E_K(p_i \oplus c_{i-1})$, with c_0=IV
  - IV is the initialization vector, is not secret, usually chosen as a random bitstring
- Decryption: $p_i = D_K(c_i) \oplus c_{i-1}$, with c_0=IV



CBC mode encryption

Properties of CBC mode

☺ By changing the IV, identical plaintexts are encrypted into different ciphertexts

- Randomized encryption
- IV must be unpredictable, it is usually chosen at random
- IV does not need to be secret, but the integrity of IV must be protected

☺ A ciphertext block depends on all preceding plaintext blocks

- Reordering of ciphertext blocks affects decryption

☹ Errors in one ciphertext block affect decryption of two blocks

- Errors in c_i affects decryption of c_i and c_{i+1}

☹ Encryption cannot be parallelized

- What about decryption?

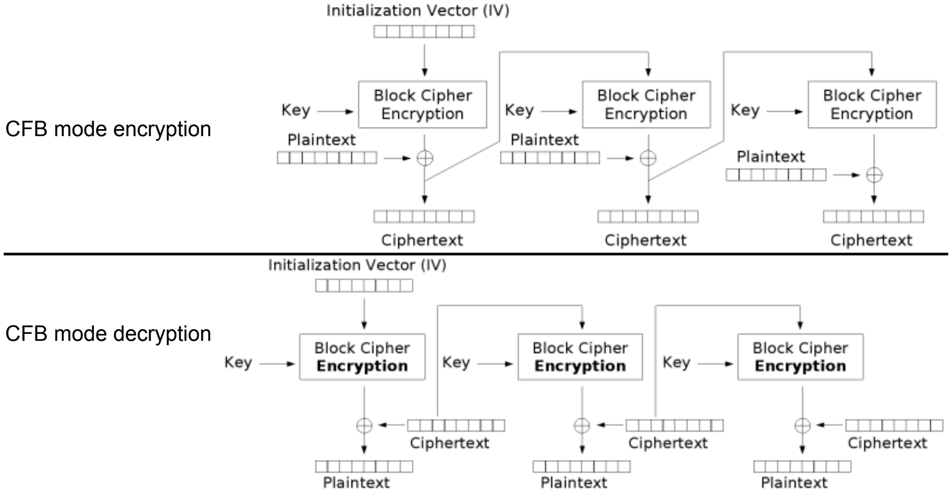
Use Block ciphers to construct Stream Ciphers

- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter Mode (CTR)
- Common properties:
  - Generate a keystream which is then XOR-ed with the plaintext
  - Uses only the encryption function of the cipher both for encryption and for decryption

- Encrypt: C = P ⊕ KEYSTREAM
  - How do you decrypt?
- Decrypt: P = C ⊕ KEYSTREAM

Cipher Feedback (CFB)

- Encryption: $c_i = E_K(c_{i-1}) \oplus p_i$, with c_0=IV, for $1 \le i \le m$
- Decryption: $p_i = E_K(c_{i-1}) \oplus c_i$, with c_0=IV, for $1 \le i \le m$



Properties of CFB mode

- Very similar with CBC mode

☺ By changing the IV, identical plaintexts are encrypted into different ciphertexts (randomized encryption)

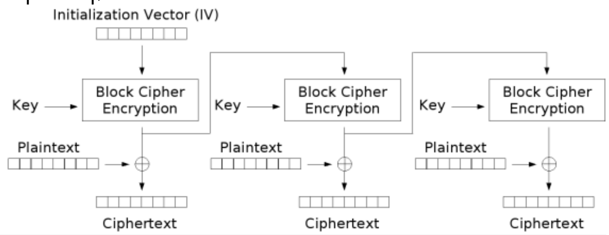
☺ A ciphertext block depends on all preceding plaintext blocks

  - Reordering of ciphertext blocks affects decryption

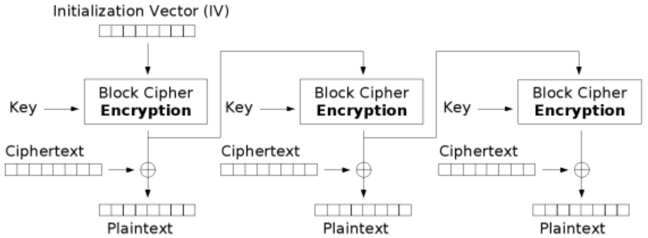
Output Feedback (OFB)

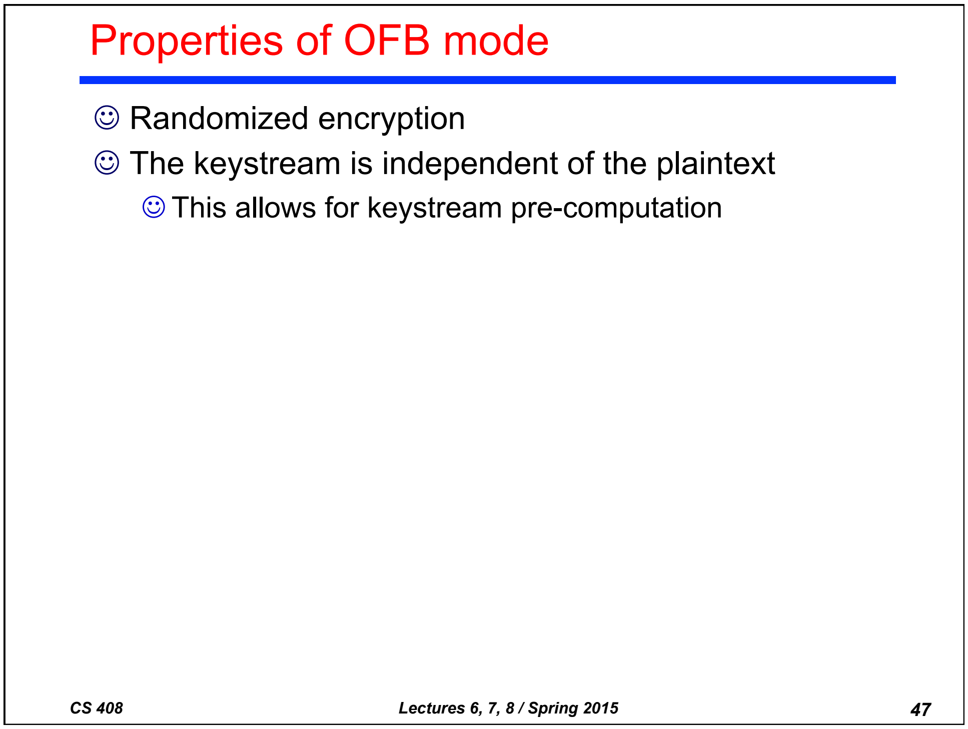
- Let $o_i = E_K(o_{i-1})$, with o_o=IV (this is called "keystream")
- Encryption: $c_i = p_i \oplus o_i$, for $1 \leq i \leq m$
- Decryption: $p_i = c_i \oplus o_i$, for $1 \leq i \leq m$

OFB mode encryption



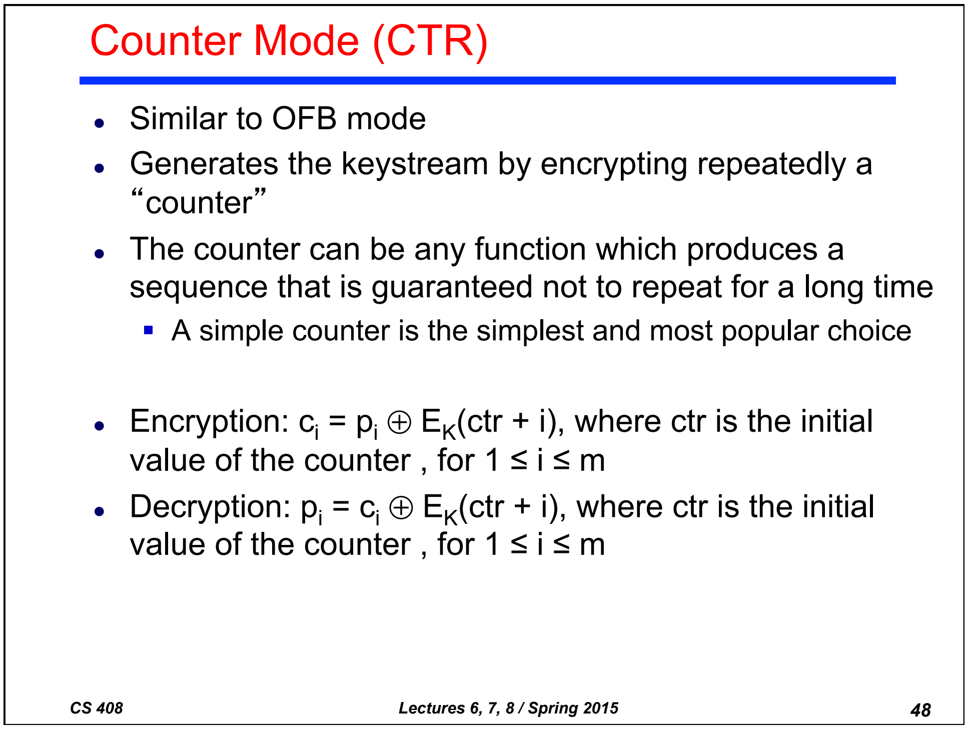
OFB mode decryption





Properties of OFB mode

- ☺ Randomized encryption
- ☺ The keystream is independent of the plaintext
  - ☺ This allows for keystream pre-computation



Counter Mode (CTR)

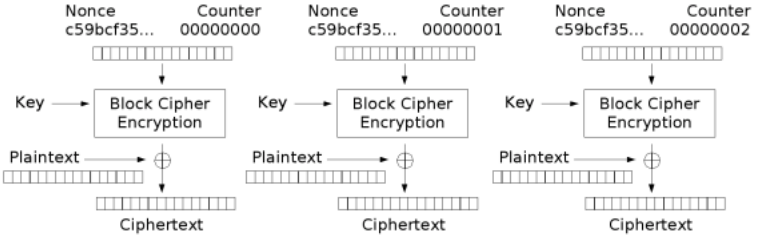
- Similar to OFB mode
- Generates the keystream by encrypting repeatedly a "counter"
- The counter can be any function which produces a sequence that is guaranteed not to repeat for a long time
  - A simple counter is the simplest and most popular choice

- Encryption: $c_i = p_i \oplus E_K(ctr + i)$, where ctr is the initial value of the counter , for $1 \leq i \leq m$
- Decryption: $p_i = c_i \oplus E_K(ctr + i)$, where ctr is the initial value of the counter , for $1 \leq i \leq m$

Counter Mode (CTR)

CTR mode

encryption


Nonce c59bcf35... Counter 00000000 | Nonce c59bcf35... Counter 00000001 | Nonce c59bcf35... Counter 00000002

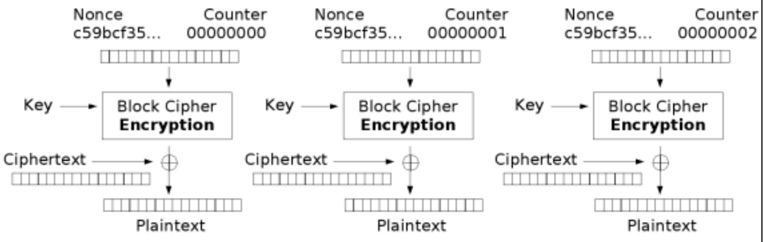
Key → Block Cipher Encryption

Plaintext → ⊕

Ciphertext

CTR mode

decryption


Nonce c59bcf35... Counter 00000000 | Nonce c59bcf35... Counter 00000001 | Nonce c59bcf35... Counter 00000002

Key → Block Cipher **Encryption**

Ciphertext → ⊕

Plaintext

Properties of CTR

- ☺ Software and hardware efficiency: different blocks can be encrypted in parallel.
- ☺ Preprocessing: the encryption part can be done offline and when the message is known, just do the XOR.
- ☺ Random access: decryption of a block can be done in random order, very useful for hard-disk encryption.
- ☺ Messages of arbitrary length: ciphertext is the same length with the plaintext (i.e., no IV).

Recommended Reading

- Chapter 4