L3: Basic Cryptography II

Hui Chen, Ph.D.
Dept. of Engineering & Computer Science
Virginia State University
Petersburg, VA 23806

Acknowledgement

- Many slides are from or are revised from the slides of the author of the textbook
  - Matt Bishop, Introduction to Computer Security, Addison-Wesley Professional, October, 2004, ISBN-13: 978-0-321-24774-5. Introduction to Computer Security @ VSU's Safari Book Online subscription
  - http://nob.cs.ucdavis.edu/book/book-intro/slides/

Overview

- Classical Cryptography
  - Caesar cipher
  - Vigènere cipher
  - DES
  - AES
- Public Key Cryptography
  - Diffie-Hellman
  - RSA
- Cryptographic Checksums
  - HMAC

Previous lecture

This and future lectures

The Data Encryption Standard

- DES = The Data Encryption Standard
  - A Product Cipher: uses both transposition and substitution
- In 1977 the National Bureau of Standards announced a Data Encryption Standard to be used in unclassified U.S. Government applications
  - For sensitive but unclassified U.S. government data
  - Unclassified U.S. Government data: information not concerned with national security
  - In wide international use
    - e.g., banks used it for funds transfer security

DES: A Block Cipher

- Input, output, and key are each 64 bits long
  - divides data into 64-bit blocks
  - uses a 64 bit key (i.e., a *key block*) supplied by user
  - encrypts the 64-bit blocks of data
  - outputs 64-bit blocks of ciphertext

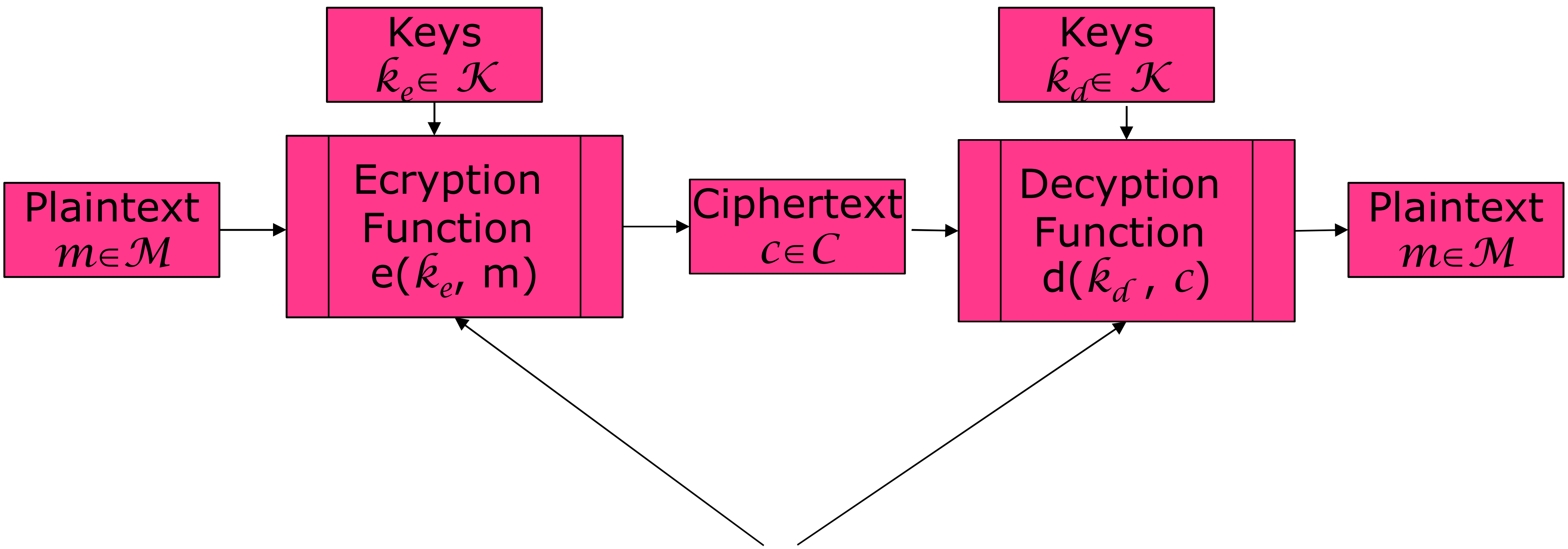
DES Key Block

- 64 bit key block, supplied by user
  - 8 bytes
  - Each byte
    - 7 bits + 1 parity bit
- 56 bit key
  - $8 \times 7 = 56$ bits
  - Drop 8 parity bits

DES Rounds

- The DES block cipher consists of 16 rounds (iterations)
  - each round with a round key generated from the user-supplied key
  - basic unit is the bit
  - each round is a product cipher, i.e., each round performs both substitution and transposition (permutation) on the bits
- The rounds are executed sequentially
  - The input of round *i+1* is the output of round *i*

Overview of DES



- 3 major steps in both encipherment and decipherment
- $k_e = k_d$

Encipherment & Decipherment

- Apply an initial permutation (IP) to the input block

  $(L_0, R_0) \leftarrow IP\ (Input\ Block)$

- Iterate 16 rounds

  $L_i \leftarrow R_{i-1}$

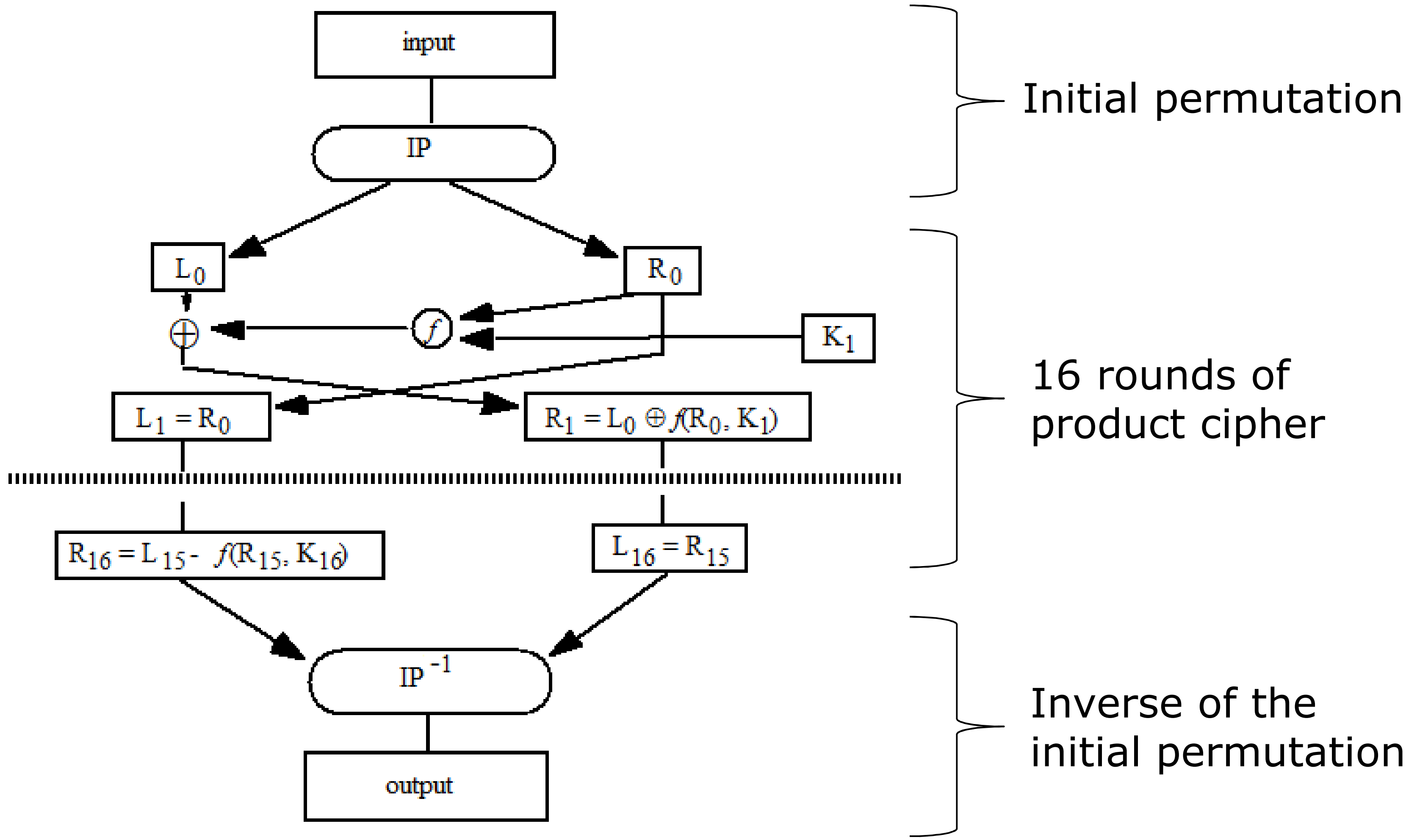
  $R_i \leftarrow L_{i-1} \oplus f(R_{i-1}, K_i)$

  - K_i is a round key, a substring of the 56-bit input key
  - *f is called S-Box function: f provides the strength of DES*

- Apply the inverse of IP to the output of round 16

  $Output\ Block \leftarrow IP^{-1}\ (R_{16}, L_{16})$

Encipherment & Decipherment



Initial & Final Permutations

- See Schneier, 1996 for more information
- Designed to load plaintext and ciphertext data into a DES chip in byte-sized pieces
- Does not affect DES's strength
- Bit-wise permutation trivial in hardware, but difficult (*inefficient*) in software
  - Many software implementations leave the input & final permutations out (they should *not* be called DES though)

IP and its Inverse

- Initial Permutation and its inverse (from Denning, 1982)

TABLE 2.3(a) Initial permutation IP.

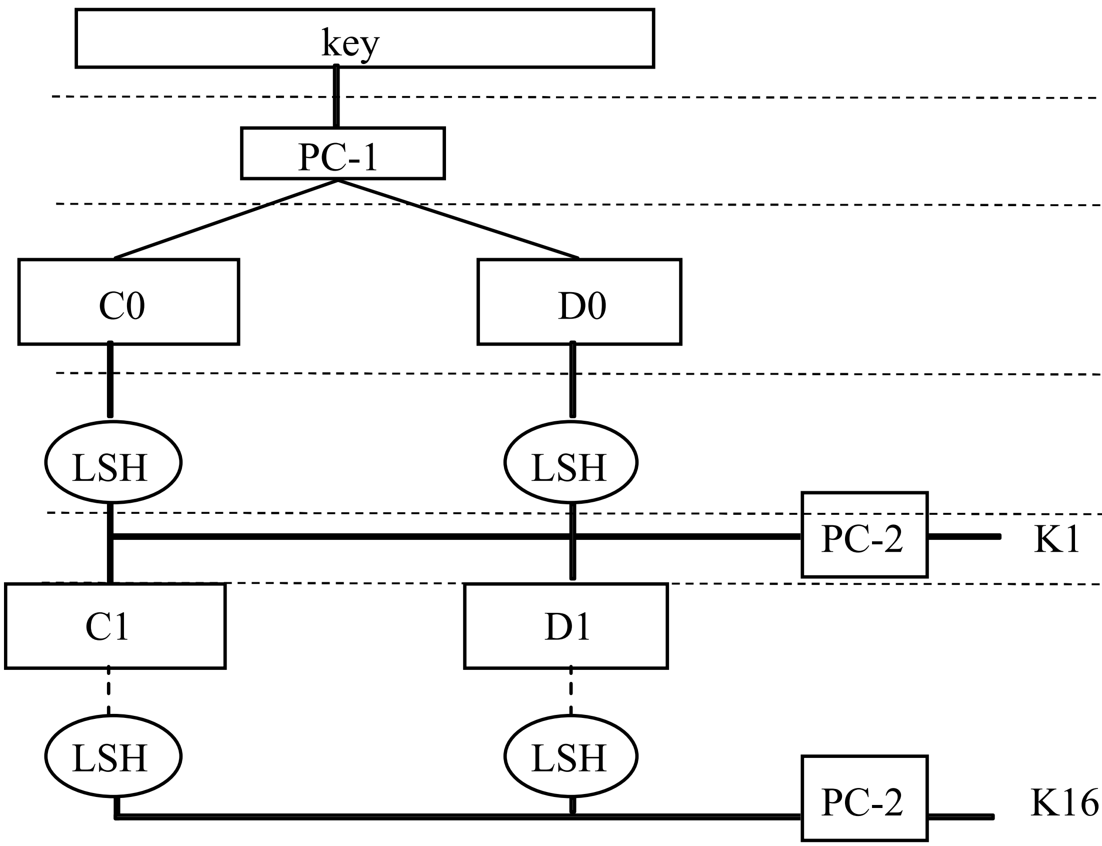
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

TABLE 2.3(b) Final 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

Generation of Round Keys

- Round keys are 48 bits each



56 bits (drop parity bits from the 64 bit key block)

A permutation (next slide)

Two 28-bit halves

Apply left circular shift
C1 = LS1(C0)   D1=LS1(D1)

K1 = PC-2(C1 D1)

Repeat for next round key

Note: PC-2 (C1 D1) takes 48 bits from (C1 D1) after a permutation

Generation of Round Keys: Permutations

- PC-1 and PC-2 are two permutations (from Denning 1982)

TABLE 2.7 Key permutation 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

PC-1: 56-bit input and output

TABLE 2.9 Key permutation 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

PC-2: 56-bit input and 48-bit output

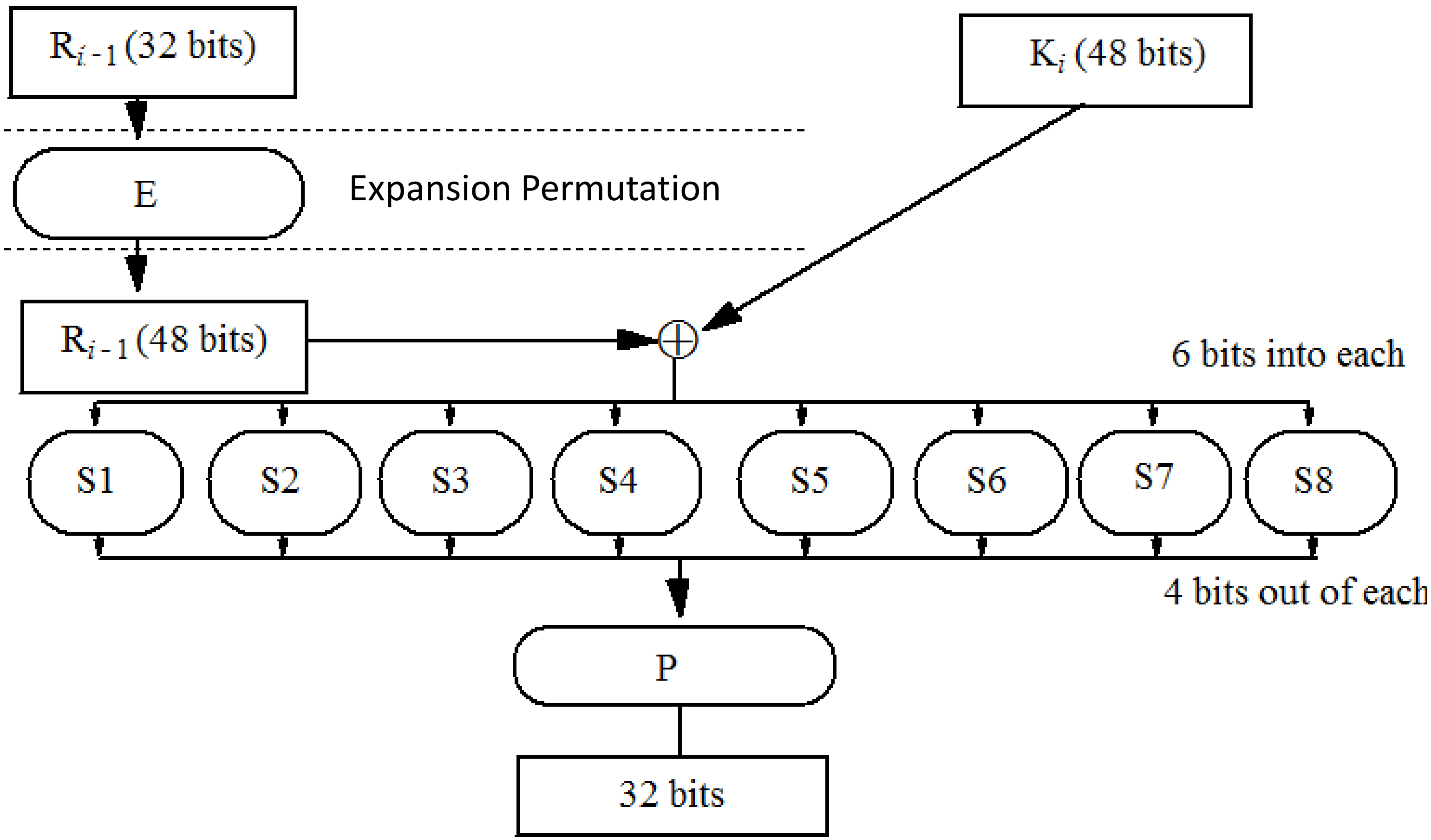
Generation of Round Keys: Left Circular Shift

- From Denning, 1982

TABLE 2.8 Key schedule of left shifts LS.

Iteration i	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

f function



R$_{i-1}$ (32 bits)
K$_i$ (48 bits)
E
Expansion Permutation
R$_{i-1}$ (48 bits)
6 bits into each
S1
S2
S3
S4
S5
S6
S7
S8
4 bits out of each
P
32 bits

Inside *f* function: Expansion Permutation

- From Schneier, 1996
  - Repeating some bits to achieve *avalanche effect*, i.e., to have every bit of the ciphertext depend on every bit of the plaintext and every bit of the key as quickly as possible.

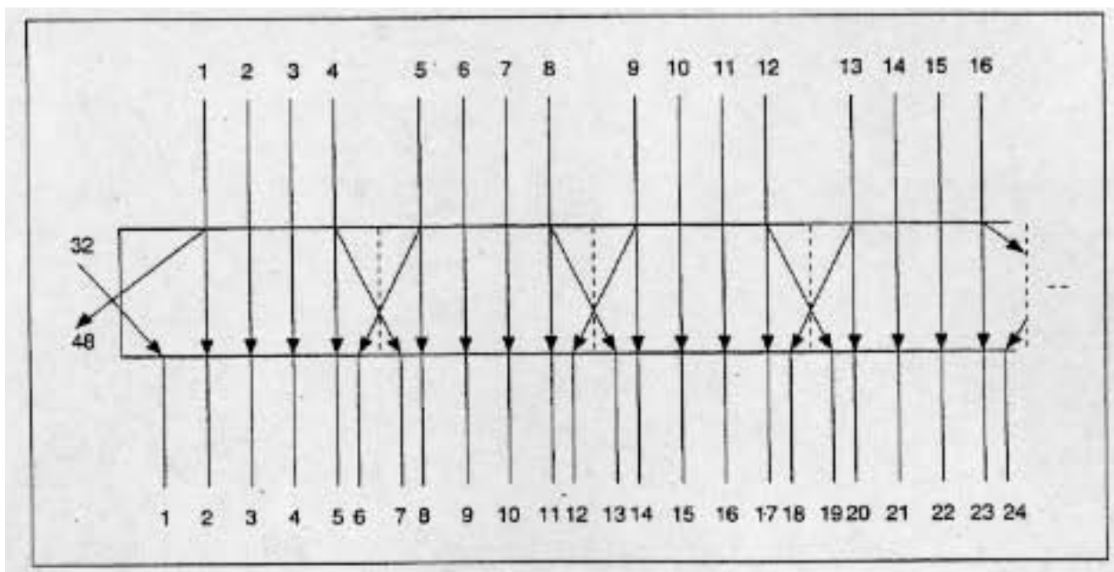


Figure 12.3 Expansion permutation.

Inside *f* function: Substitution Boxes

- S-Boxes: From Denning, 1982 (and for complete table)
- 6 bit input

  b1b2b3b4b5b6

  b1b6 selects row

  b2b3b4b5 selects column
- Example

  Input: $(010011)_2$

  b1b6 = $(01)_2 = 1$

  b2b3b4b5 = $(1001)_2 = 9$

  Select 6 = $(0110)_2$

TABLE 2.6 Selection functions (S-boxes).

	Column																
Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S_1
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S_2
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	S_3
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	

Inside *f* function: P-Box Permutation

- From Schneier, 1996

Table 12.7
P-Box Permutation

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

Controversy

- Diffie and Hellman claim that in a few years technology would allow DES to be broken in days (Diffie and Hellman, 1977)
- Design of efficient attacks using 1999 technology published
  - See “Chronology” in *https://en.wikipedia.org/wiki/Data_Encryption_Standard*
- Design decisions of S-boxes not public
  - S-boxes may have backdoors

Undesirable Properties

- 4 weak keys
  - They are their own inverses
- 12 semi-weak keys
  - Each has another semi-weak key as inverse
- Complementation property
  - $DES_k(m) = c \Rightarrow DES_{k'}(m') = c'$
- S-boxes exhibit irregular properties
  - Distribution of odd, even numbers non-random
  - Outputs of fourth box depends on input to third box

Note: DES key space:
2^{56} = 72,057,594,037,927,936 keys

Choosing weak keys (very unlikely) leads to the same round keys

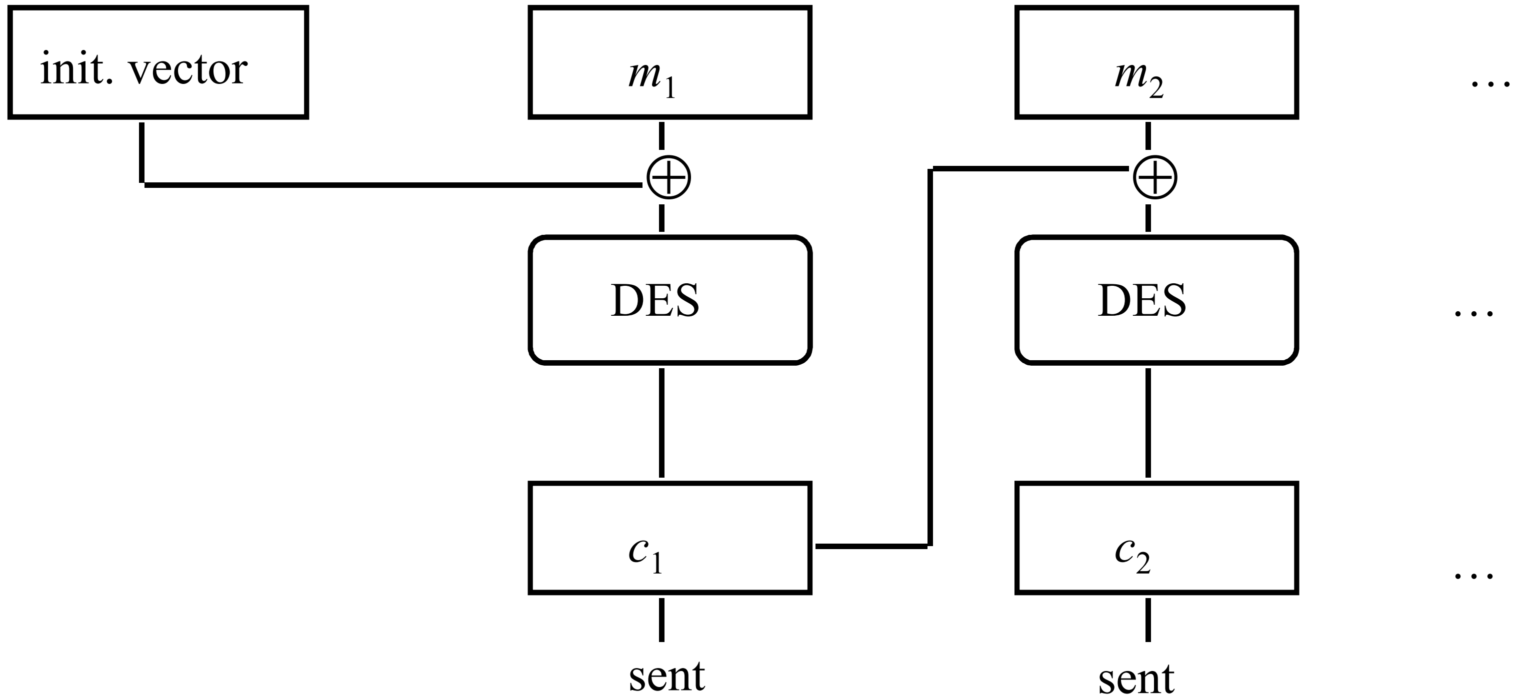
Differential and Linear Cryptanalysis on DES

- Chosen ciphertext attacks
- Differential cryptanalysis: based on how differences in inputs correlate with difference in outputs
  - Requires 2^{47} plaintext-ciphertext pairs
  - Revealed several properties
    - Small changes in S-boxes reduce the number of pairs needed
    - Making every bit of the round keys independent does not impede attack
- Linear cryptanalysis: based on correlations between inputs and outputs
  - improved result, requires 2^{43} plaintext-ciphertext pairs

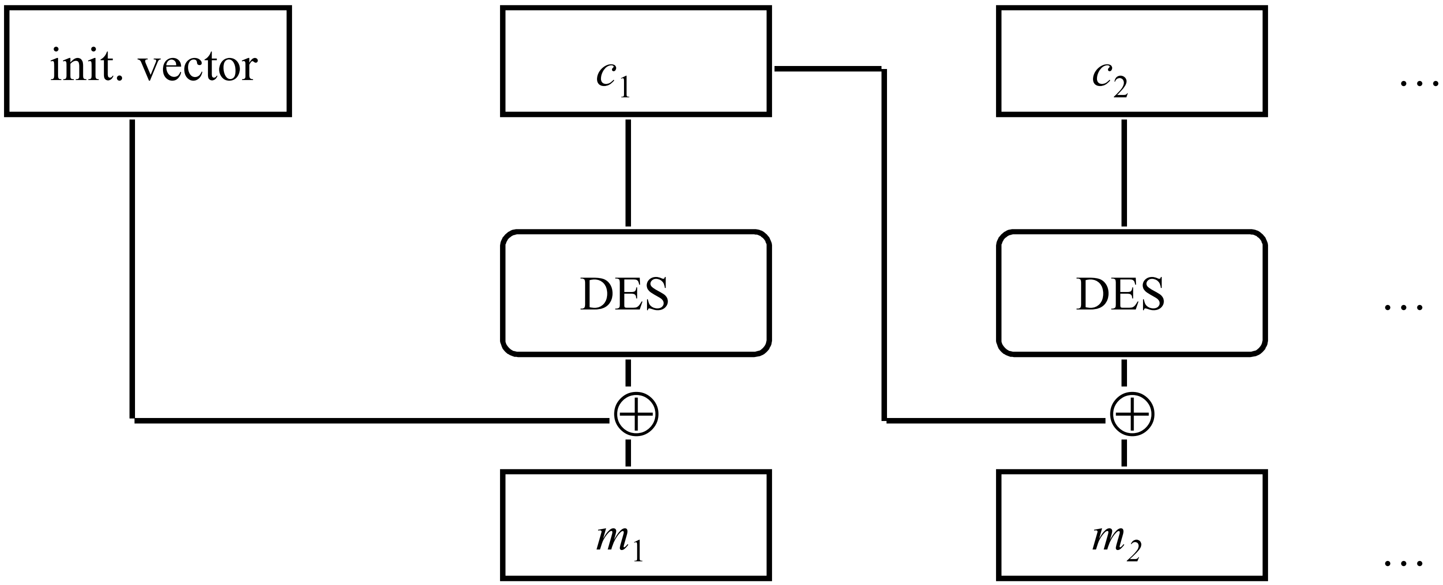
DES Modes

- Electronic Code Book Mode (ECB)
  - Encipher each block independently
- Cipher Block Chaining Mode (CBC)
  - Xor each block with previous ciphertext block
  - Requires an initialization vector for the first one
- Encrypt-Decrypt-Encrypt Mode (2 keys: k, k')
  - $c = DES_k(DES_{k'}^{-1}(DES_k(m)))$
- Encrypt-Encrypt-Encrypt Mode (3 keys: k, k', k'')
  - $c = DES_k(DES_{k'}(DES_{k''}(m)))$

CBC Mode Encryption



CBC Mode Encryption



Self-Healing Property

- Initial message
  - `3231343336353837 3231343336353837`
    `3231343336353837 3231343336353837`
- Received as (underlined 4c should be 4b)
  - `ef7c4cb2b4ce6f3b f6266e3a97af0e2c`
    `746ab9a6308f4256 33e60b451b09603d`
- Which decrypts to
  - `efca61e19f4836f1 3231333336353837`
    `3231343336353837 3231343336353837`
  - Incorrect bytes underlined
  - Plaintext "heals" after 2 blocks

Current Status of DES

- Design for computer system, associated software that could break any DES-enciphered message in a few days published in 1998
- Several challenges to break DES messages solved using distributed computing
- NIST selected the *Rijndael* cipher as *Advanced Encryption Standard*, successor to DES
  - Designed to withstand attacks that were successful on DES

AES: Result of Open Competition

- NIST held an *open* competition and selected the *Rijndael* cipher as Advanced Encryption Standard (AES), a successor to DES
  - NIST issued call for AES cipher in 1997 (http://csrc.nist.gov/archive/aes/pre-round1/aes_9709.htm)
  - 15 candidates accepted in June 1998 (http://csrc.nist.gov/archive/aes/round1/r1report.htm)
  - 5 finalists announced in August 1999 (http://csrc.nist.gov/archive/aes/round1/r1report.htm)
  - *Rijndael* cipher accepted as the winner and AES (http://www.nist.gov/public_affairs/releases/g00-176.cfm and http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf)
  - Designed by Vincent Rijmen and Joan Daemen in Belgium

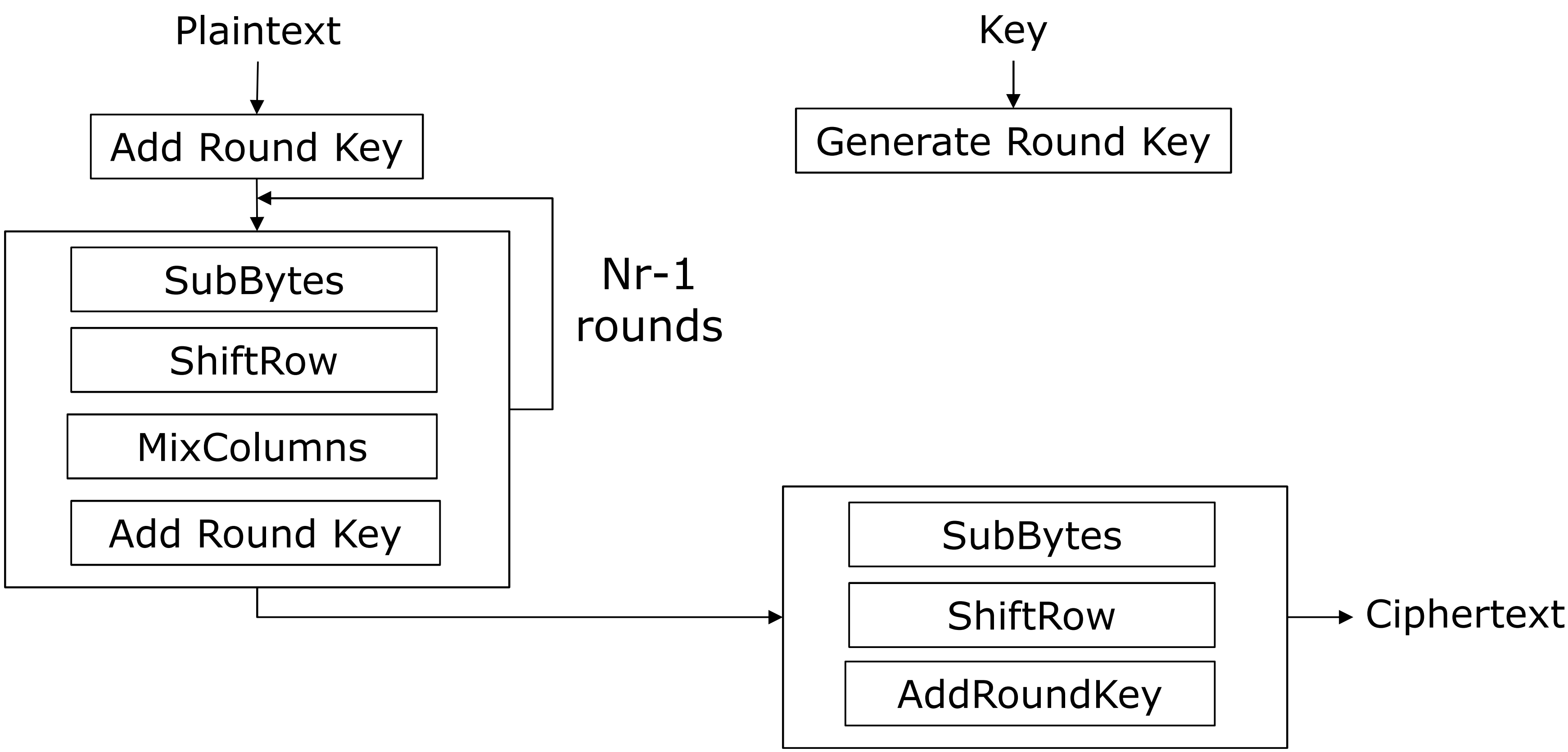
Overview

- Some similarity to DES
  - A product cipher (with transposition and substitution)
  - Operates in rounds
- AES operates on blocks of 128 bits
- AES can use keys of 128, 192, or 256 bits
- Key-block-round combination (a word = 4 bytes = 32 bits): final round slightly different from first Nr – 1 rounds

	Key Length (Nk words)	**Block Size (Nb words)**	**Number of Rounds (Nr)**
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

AES Round

- Final round slightly different from first Nr – 1 rounds



Attacks on AES

- Differential Cryptanalysis
  - High number of rounds increases difficulty of the attack
- Linear Cryptanalysis
  - AES S-box (SubBypes) and MixColumns make the attack difficult

Exercise L3-1

- Using DES and AES as examples, argue why an encryption algorithm should not contain secret design parts?
- Submit your answer in Blackboard (cite references properly if you use any)
- See the class website for the submission deadline

Exercise L3-2

- Suppose that a user chooses the keys used with DES to be only of the letters A-Z and 8 letters long. Give an approximation of the length of time it would take to try all such keys using exhaustive search, assuming each key can be tested in 1 μsec. Do the same for keys 8 letters (i.e., A-Z and a-z) or digits (i.e., 0-9) long.
- Submit your answer in Blackboard.
- See the class website for the submission deadline

Summary

- Classical cryptography in practice
  - DES
  - AES
- A few important items
  - Key and key space
  - Operation modes
  - Chosen ciphertext attacks
  - Differential cryptanalysis
  - Linear Cryptanalysis
  - Design philosophy (open or close?)