CSEN1001

Computer and Network Security

Mervat AbuElkheir
Mohamed Abdelrazik
Ahmad Helmy

Lecture (3)

Block Ciphers

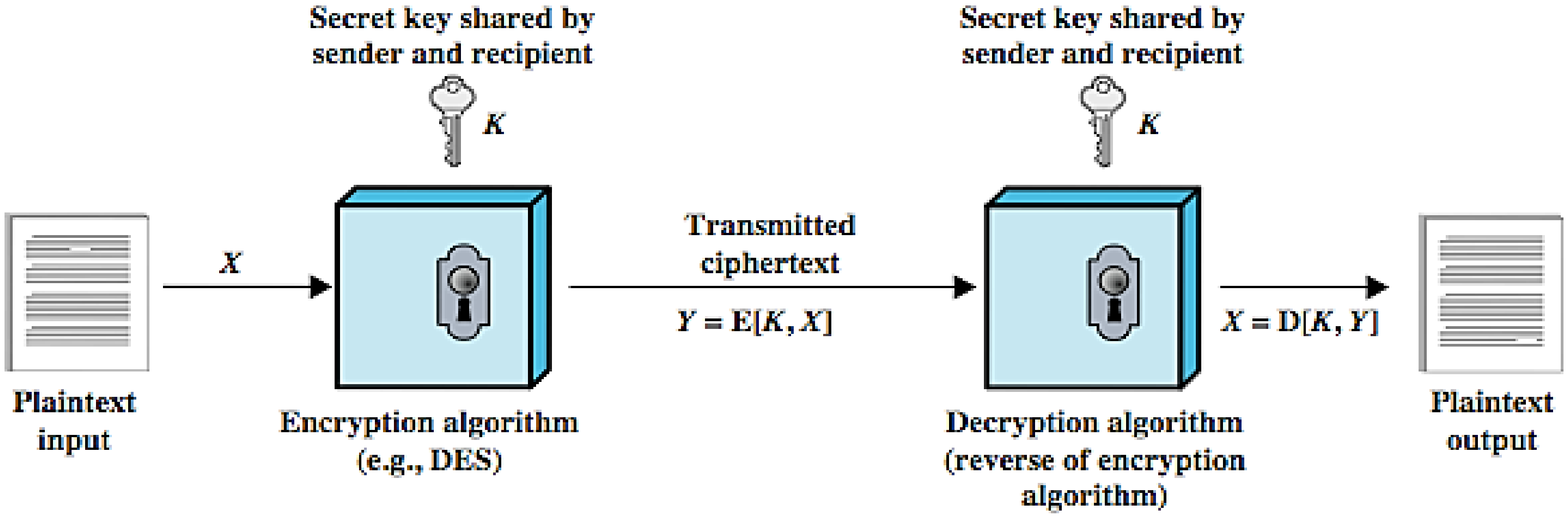
Cryptographic Tools

- Cryptographic algorithms are an important element in security services

- Review various types of elements
  - symmetric encryption
  - public-key (asymmetric) encryption
  - digital signatures and key management
  - secure hash functions
- Example is to encrypt stored data

- Characterize cryptographic system by:
  - Type of encryption operations used
    - substitution / transposition / product
  - Number of keys used
    - single-key or private / two-key or public
  - Way in which plaintext is processed
    - block / stream

Symmetric Encryption



Requirements

- Two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver
- Mathematically have:

  $Y = \mathrm{E}_K(X)$

  $X = \mathrm{D}_K(Y)$
- Assume encryption algorithm is known
- Implies a secure channel to distribute key

Attacking Symmetric Encryption

- Brute-force attack
  - try all possible keys on some ciphertext until get an intelligible translation into plaintext
- Cryptanalysis
  - rely on the nature of the algorithm
  - plus some knowledge of plaintext characteristics
  - even some sample plaintext-ciphertext pairs
  - exploits characteristics of algorithm to deduce specific plaintext or key

Cryptanalysis Attacks

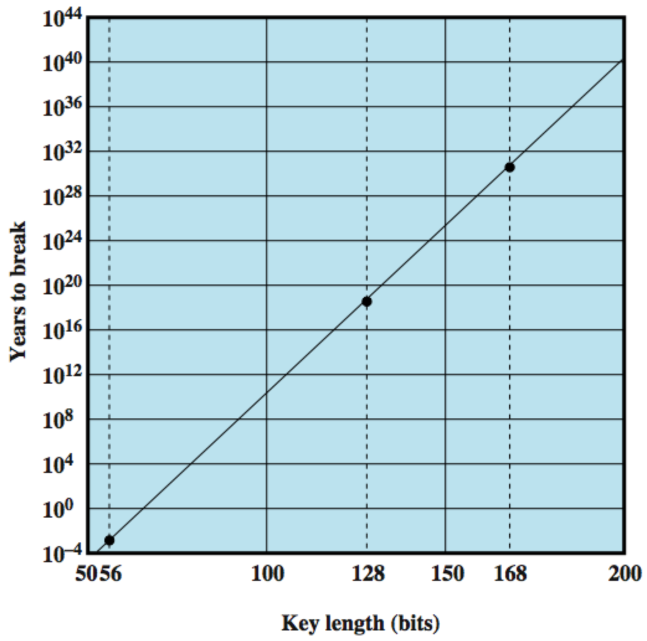
- **Ciphertext only**
  - only know algorithm & ciphertext, is statistical, know or can identify plaintext
- **Known plaintext**
  - know/suspect plaintext & ciphertext
- **Chosen plaintext**
  - select plaintext and obtain ciphertext
- **Chosen ciphertext**
  - select ciphertext and obtain plaintext
- **Chosen text**
  - select plaintext or ciphertext to en/decrypt

Encryption Schemes

- An encryption scheme is **unconditionally secure** if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available
- An encryption scheme is said to be **computationally secure** if:
  - The cost of breaking the cipher exceeds the value of the encrypted information
  - The time required to break the cipher exceeds the useful lifetime of the information

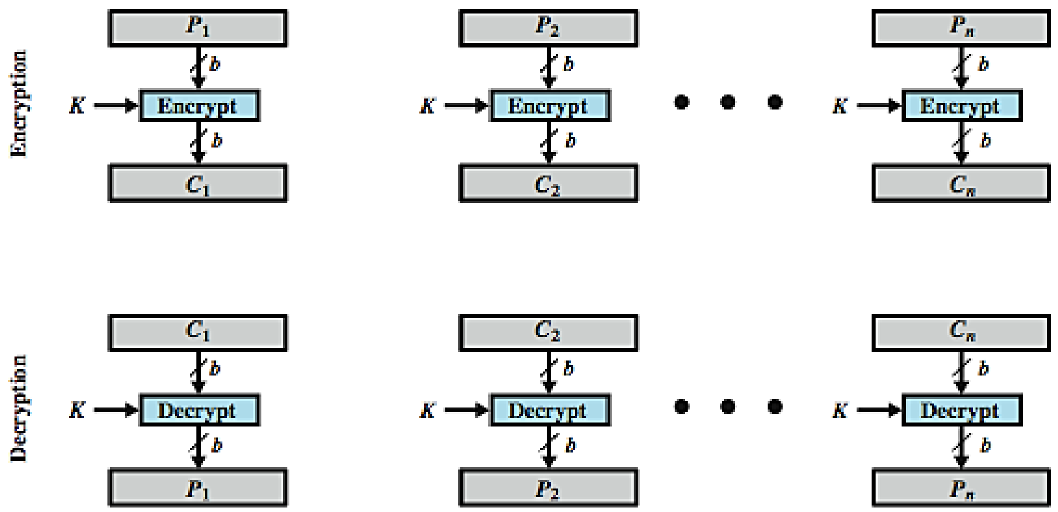
Exhaustive Key Search

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/µs	Time Required at 10^6 Decryptions/µs
32	$2^{32} = 4.3 \times 10^9$	2^{31} µs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} µs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} µs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} µs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} µs = 6.4×10^{12} years	6.4×10^6 years

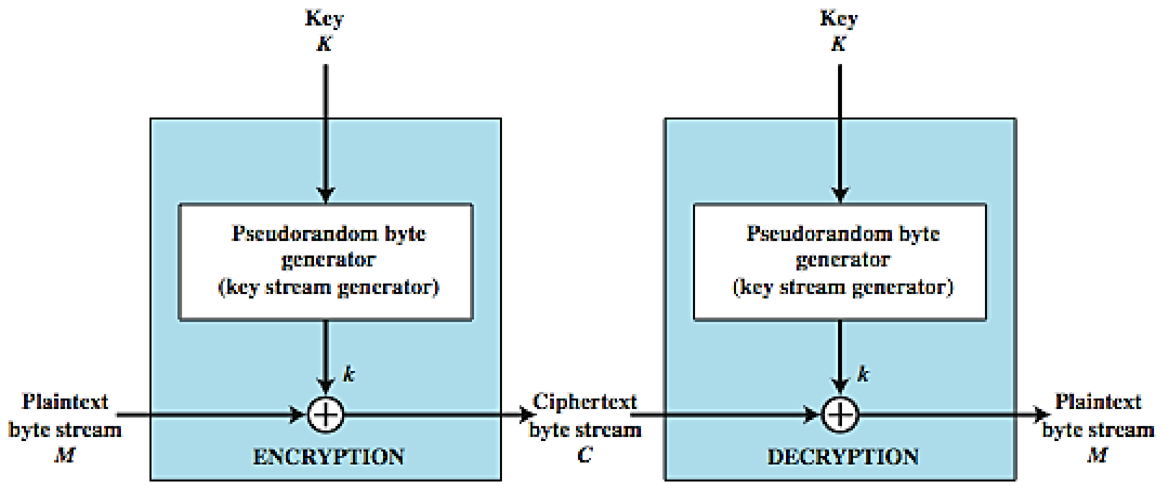


Block vs Stream Ciphers

- Block ciphers process messages in blocks, each of which is then en/decrypted
- Like a substitution on very big characters
  - 64-bits or more
- Stream ciphers process messages a bit or byte at a time when en/decrypting



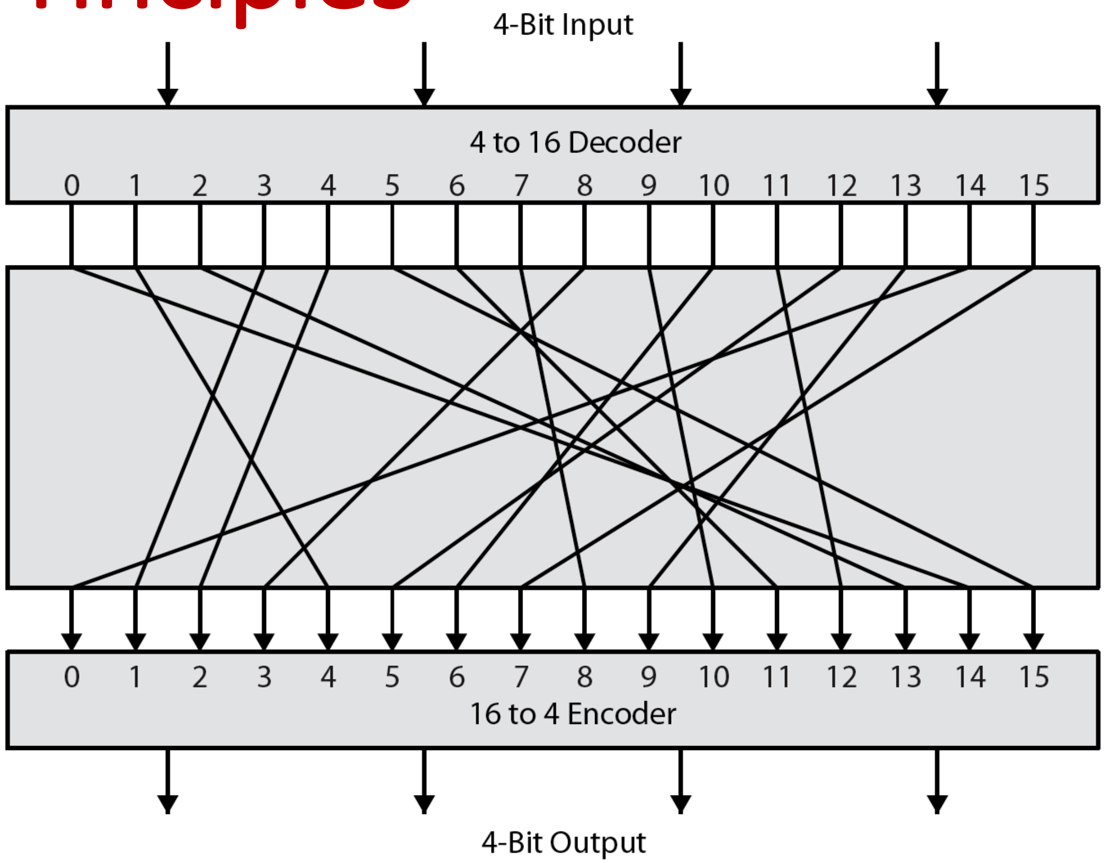
(a) Block cipher encryption (electronic codebook mode)



(b) Stream encryption

Block Cipher Principles

- Block ciphers look like an extremely large substitution
- Would need table of 2^{64} entries for a 64-bit block
- Instead create from smaller building blocks
- Most symmetric block ciphers are based on a **Feistel Cipher Structure**
- Needed since we must be able to **decrypt** ciphertext to recover messages efficiently
- Using idea of a product cipher



Ideal Block Cipher

Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

Shannon Substitution-Permutation Ciphers

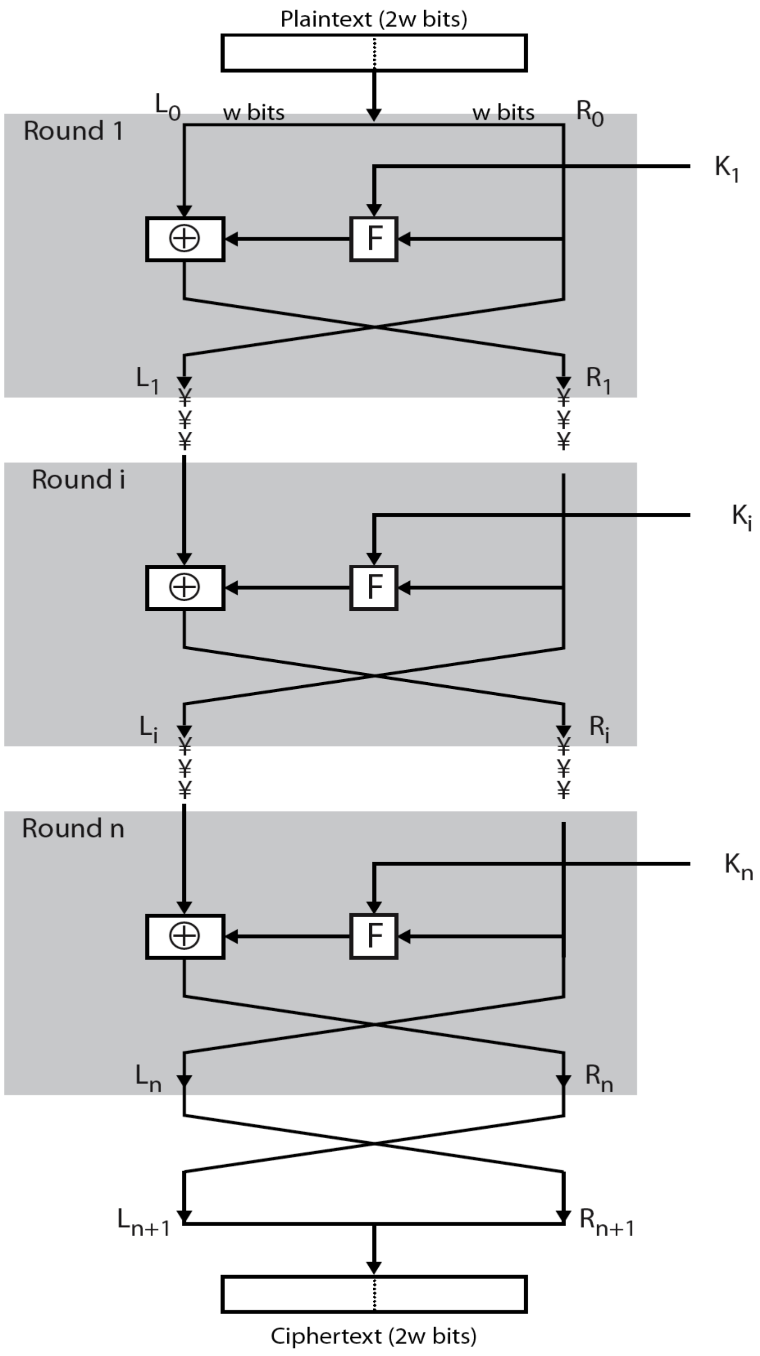
- Claude Shannon introduced the idea of substitution-permutation (S-P) networks in his 1949 paper
- Form basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations seen before:
  - *substitution* (S-box)
  - *Permutation (transposition)* (P-box)
- Provide ***confusion*** & ***diffusion*** of message & key

Confusion and Diffusion

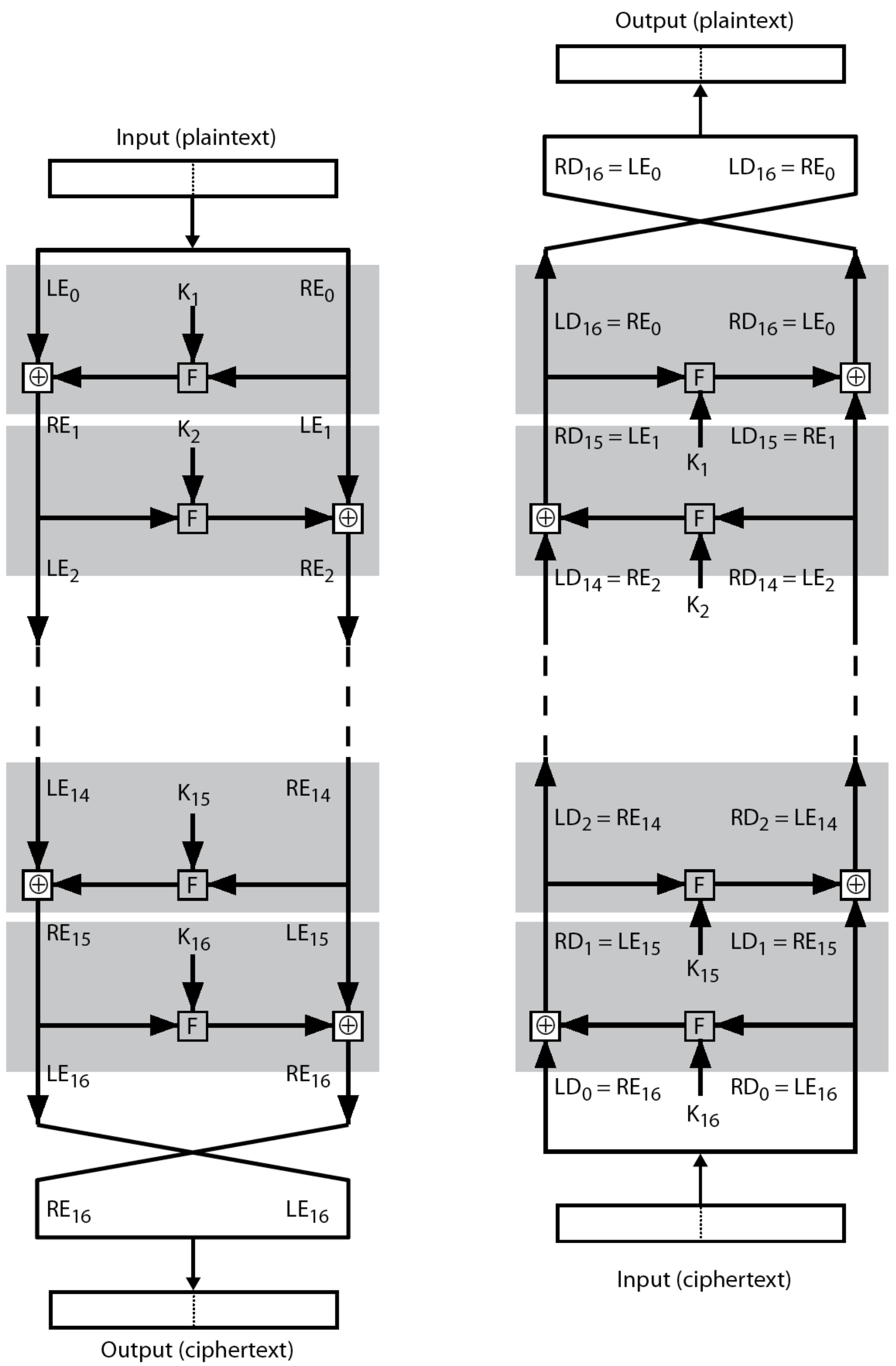
- Cipher needs to completely obscure statistical properties of original message
- A one-time pad does this
- More practically Shannon suggested combining S & P elements to obtain:
  - **Confusion** – make relationship between ciphertext and key as complex as possible
  - **Diffusion** – dissipate statistical structure of plaintext over bulk of ciphertext

Feistel Cipher

- Horst Feistel devised the **Feistel cipher**
  - based on the concept of **invertible product cipher**
- Partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- implements Shannon's S-P net concept
- Design elements of Feistel cipher include:
  - block size
  - key size
  - number of rounds
  - subkey generation algorithm
  - round function
  - fast software en/decryption
  - ease of analysis



Feistel Cipher Decryption



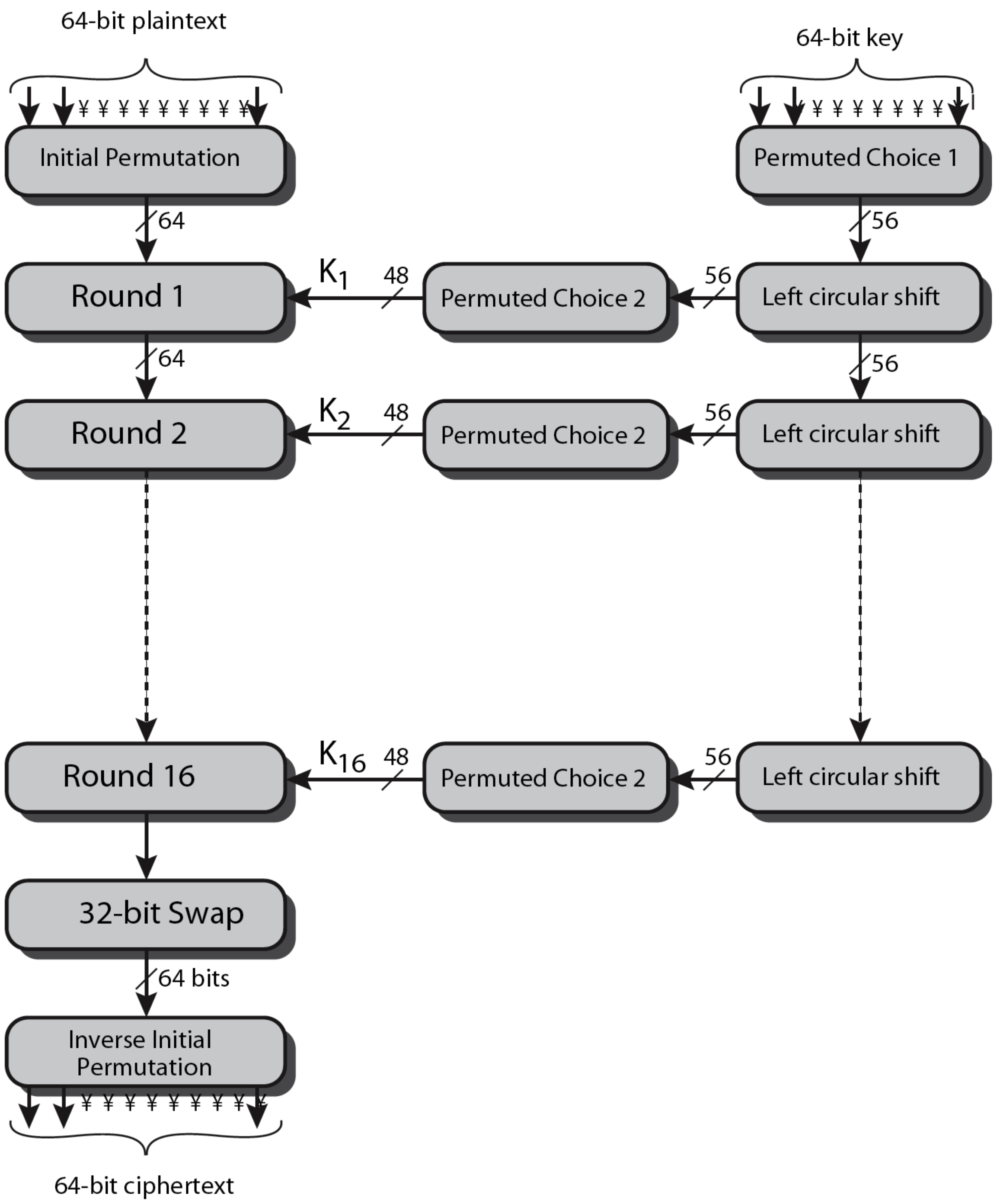
Data Encryption Standard (DES)

- Was most widely used block cipher in world
- Adopted in 1977 by NIST (formerly NBS)
  - as FIPS 46
- Encrypts 64-bit data using 56-bit key
- Has widespread use

- IBM developed Lucifer cipher
  - by team led by Feistel in late 60's
  - used 64-bit data blocks with 128-bit key
- Then redeveloped as a commercial cipher with input from NSA and others

- In 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES
- Although DES standard is public, there was considerable controversy over design
  - in choice of 56-bit key (vs Lucifer 128-bit)
  - and because design criteria were classified
- Subsequent events and public analysis show in fact design was appropriate
- Use of DES flourished
  - especially in financial applications
  - still standardized for legacy application use

DES Encryption



Initial Permutation (IP)

- First step of the data computation
- IP reorders the input data bits

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

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

(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})

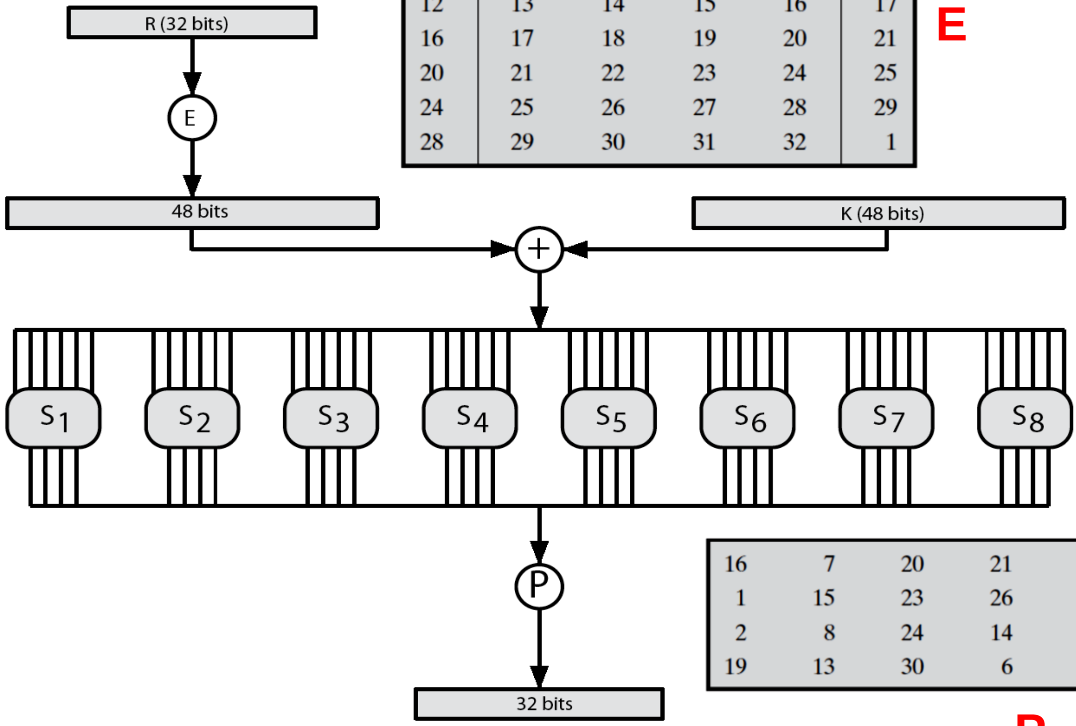
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

DES Round Structure

- Uses two 32-bit L & R halves
- As for any Feistel cipher can describe as:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

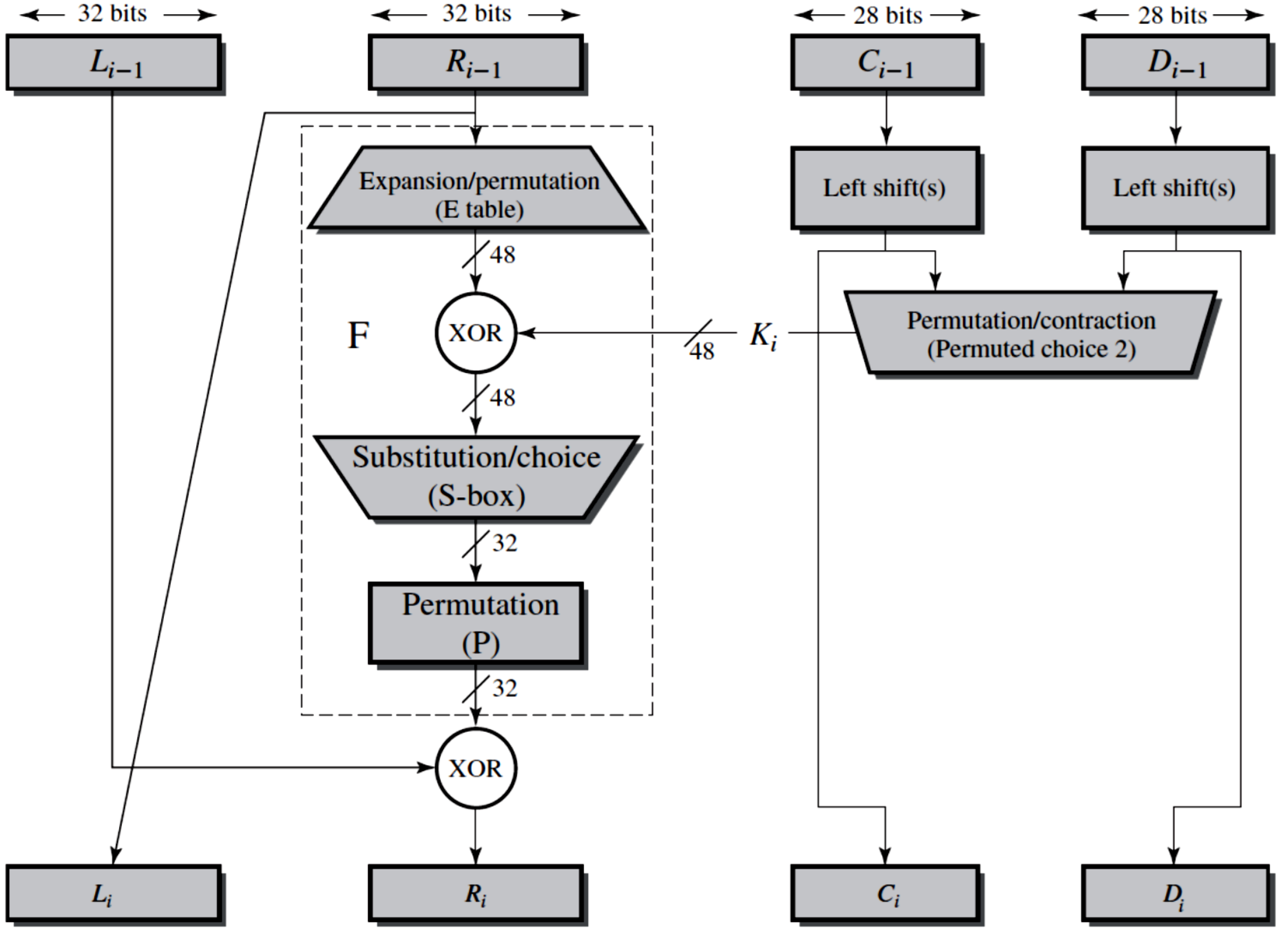
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

E



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

P



- F takes 32-bit R half and 48-bit subkey:
  - expands R to 48-bits using perm E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes using 32-bit perm P

Substitution S-Boxes

- Eight S-boxes which map 6 to 4 bits
- Each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (**row** bits) of input select one row of 4
  - inner bits 2-5 (**col** bits) of input are substituted
  - result is 8 lots of 4 bits, or 32 bits
- Row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- Example:

```
If the input to S1 is 011001, the
output is row 1 col 12 → 9 = 1001
```

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
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

Key Schedule

- Forms subkeys used in each round
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K

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

    - selecting 24-bits from each half & permuting them by PC2 for use in round function F
- Note practical use issues in h/w vs s/w

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

64-bit Key

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

PC1 – 56 bits (IP)

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

PC2 – 48 bits

DES Decryption

- Decryption must unwind steps of data computation
- With Feistel design, do encryption steps again using subkeys in reverse order (SK16 … SK1)
  - IP undoes final FP step of encryption
  - 1st round with SK16 undoes 16th encrypt round
  - ….
  - 16th round with SK1 undoes 1st encrypt round
  - then final FP undoes initial encryption IP
  - thus recovering original data value

The contents of this lecture can be found in Ch3 of “Cryptography and Network Security”