Symmetric Key Cryptography

Haipeng Dai

haipengdai@nju.edu.cn
313 CS Building
Department of Computer Science and Technology
Nanjing University

Basic Terms

- **Threat, vulnerability, attack, and intrusion**
- **Threat**: attackers, angry employees, etc.
- **Vulnerability**: weakness of a system
- **Attack**: actions to make harm to a system by **modifying** the system, **reading** information from the system, or **stopping** the system from serving its legitimate users
  - Passive attacks: read information in a system
    - e.g., Eavesdropping
  - Active attacks: modify a system
    - e.g., message modification, insertion, deletion, replay
- **Intrusion**: successfully modifying a system or reading information from the system

Seven Security Properties

- Authentication
- Confidentiality
- Integrity
- Non-repudiation
- Authorization
- Freshness
- Availability

Security Property 1: Authentication

- **Authentication** (authenticity)
  - Verify an identity claimed to be
  - Mechanisms:
    - Something the user **is**
      - e.g., fingerprint or retinal pattern, DNA sequence, unique bio-electric signals produced by the living body, or other biometric identifier
    - Something the user **has**
      - e.g., ID card, security token, software token or cell phone
    - Something the user **knows**
      - e.g., a password, a pass phrase or a personal identification number (PIN)
    - Something the user **does**
      - e.g., voice recognition, signature, or gait

Security Property 2: Confidentiality

- **Confidentiality** (secrecy)
  - Protect information from leaking.
  - Two types:
    - Message content confidentiality
    - Message header confidentiality: who talks to whom is secret.
  - Mechanisms
    - Encryption
    - Traffic padding

Security Property 3: Integrity

- **Integrity**
  - Protect system/data from being modified.
  - System integrity
    - Prevent modification to system
      - e.g., communication system: message modification, insertion, deletion, and replay (integrity of communication channels)
  - Data integrity
    - Prevent modification to data
      - e.g., communication system: message modification
  - Mechanisms:
    - Message Digest

Security Property 4: Non-repudiation

- **Non-repudiation**
  - Prevent someone from denying their action.
    - E.g., creating a message.
  - Mechanisms:
    - Message Digest

Security Property 5: Authorization

- **Authorization**
  - Give someone permission to do something (such as access a resource) and enforce that they don't do anything beyond their permission
  - Mechanisms:
    - Access Control

Security Property 6: Freshness

- **Freshness**
  - Verify that message is recent, is not replayed
  - e.g., a check becomes invalid if not cashed within 6 months
    - The expired check still has integrity, but not freshness.
  - Mechanisms
    - Nonce
    - Expiration time

Security Property 7: Availability

- **Availability**:
  - Keep service available to legitimate users
  - Deny of Service attacks

Basic Cryptography Terminology

- plaintext - the original message
- ciphertext - the coded message
- cipher - algorithm for transforming plaintext to ciphertext
- key - info used in cipher known only to sender/receiver
- encipher (encrypt) - converting plaintext to ciphertext
- decipher (decrypt) - converting ciphertext to plaintext
- cryptography - study of encryption principles/methods
- cryptanalysis (codebreaking) - the study of principles/ methods of deciphering ciphertext without knowing key
- cryptology - the field of both cryptography and cryptanalysis

Cryptography Primitives: Encryption/Decryption function

- Symmetric Keys

- Main Properties
  - Given plaintext and a key, it is computationally efficient to compute the ciphertext.
  - Given ciphertext and a key, it is computationally efficient to compute the plaintext.
  - But, given ciphertext without the key, it is computationally infeasible to compute the plaintext.
  - Keys are secret, but algorithms are public
    - Any algorithm is hard to keep secret if used widely
      - Reverse engineering, social engineering
    - Public examination helps to find flaws
    - Military keeps algorithms secret is to avoid giving enemy good ideas

Types of Attacks on Encrypted Messages

- Ciphertext only:
  - Attacker knows: only ciphertext.
- Known plaintext:
  - Attacker knows: (1) ciphertext, (2) one or more plaintext-ciphertext pairs formed with the key.
- Chosen plaintext:
  - Attacker knows: (1) ciphertext, (2) plaintext messages chosen by the attacker, together with its corresponding ciphertext generated with the key.
- Chosen ciphertext:
  - Attacker knows: (1) ciphertext, (2) purported ciphertext chosen by attacker, together with its corresponding plaintext generated with the key.
- Chosen text
  - Attacker knows: (1) ciphertext, (2) plaintext messages chosen by the attacker, together with its corresponding ciphertext generated with the key, (3) purported ciphertext chosen by attacker, together with its corresponding plaintext generated with the key.

Unconditional vs. Computational Security

- **Unconditional security**
  - No matter how much computer power is available, the cipher cannot be broken
  - The ciphertext provides insufficient information to uniquely determine the corresponding plaintext
  - Only one-time pad scheme qualifies
    - Each message has a random key of the same length
    - A simple XOR will work
    - Brute-force search all keys does not work because multiple plausible plaintext
    - Problems:
      - Key distribution and protection is difficult.
      - Making large number of random keys is a significant task.
- **Computational security**
  - The cost of breaking the cipher exceeds the value of the encrypted info
  - The time required to break the cipher exceeds the useful lifetime of the info

Brute Force Search

- Always possible to simply try every key
- Most basic attack, proportional to key size
- Assume attackers either know or recognise plaintext

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

Classic Ciphers

- We cover two examples:
  - Substitution Cipher (also called “monoalphabetic cipher”)
  - Playfair Cipher

Substitution Cipher (1/2)

- Substitute cipher is used to encrypt ordinary English text.
- The encryption and decryption rules are all permutations of alphabetic characters.
- Example: if the encryption rule is

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*
X	*N*	*Y*	*A*	*H*	*P*	*O*	*G*	*Z*	*Q*	*W*	*B*	*T*	*S*	*F*	*L*	*R*	*C*	*V*	*M*	*U*	*E*	*K*	*J*	*D*	*I*

The decryption rule is

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*
d	*l*	*r*	*y*	*v*	*o*	*h*	*e*	*z*	*x*	*w*	*p*	*t*	*b*	*g*	*f*	*j*	*q*	*n*	*m*	*u*	*s*	*k*	*a*	*c*	*i*

Plaintext: have a nice weekend

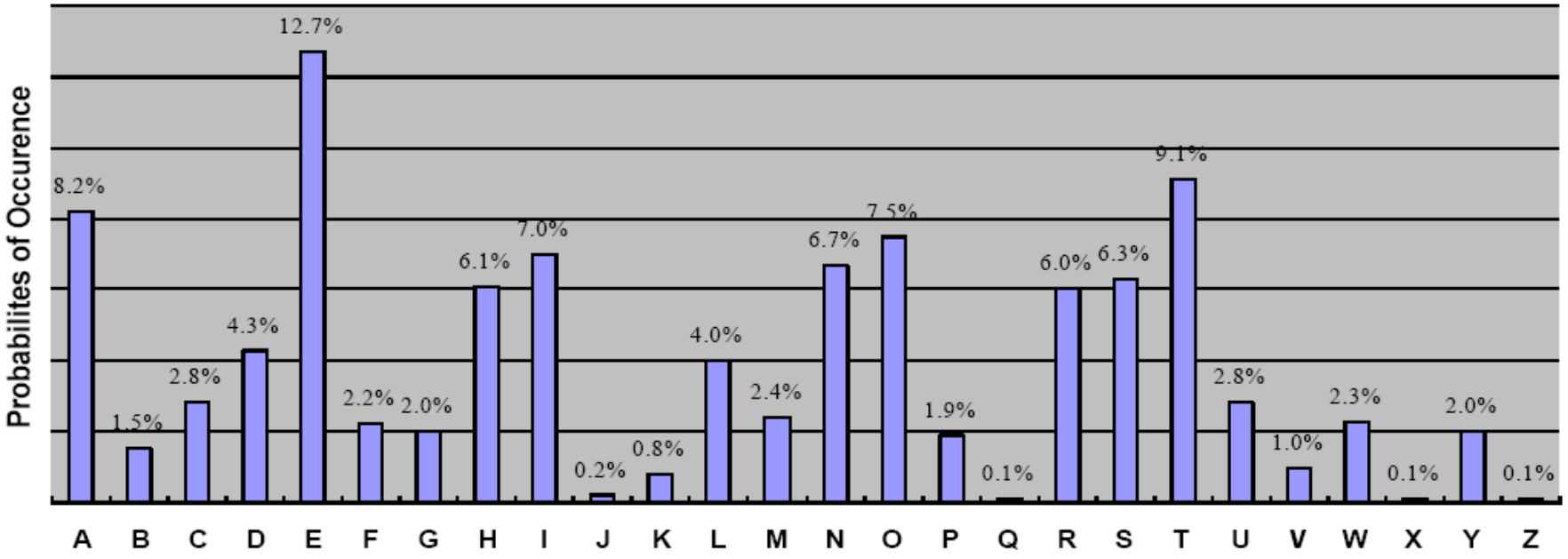
Ciphertext: GXEHXSZYHKHWHSZ

Substitution Cipher (2/2)

- Size of key space:

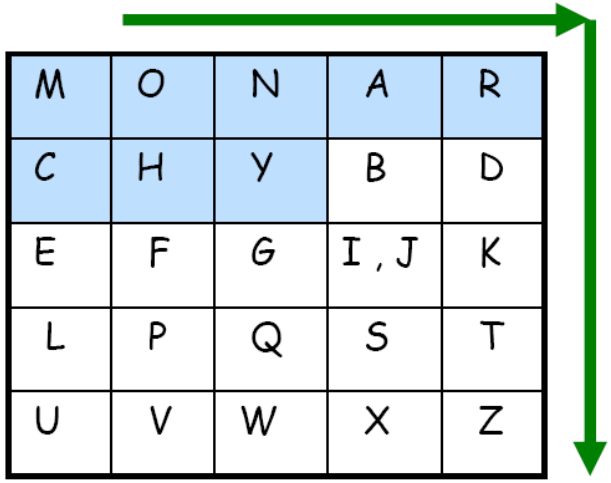
$$26! = 403291461126605635584000000 \approx 4 \times 10^{26}$$

- Still not large enough: can be broken using ciphertext only attacks.
  - Even faster: the attack uses the probabilities of occurrence of the 26 letters and their combinations.



Playfair Cipher (1/3)

- The best-known multiple-letter encryption cipher.
- The encryption is based on the use of a 5×5 matrix of letters constructed using an encryption keyword.
- Fill in letters of keywords. Skip the duplicates.
- Fill the rest of matrix with other letters in alphabetic order.
- The letters I and J count as one letter.
- Example: Keyword is **MONARCHY**



M	O	N	A	R
C	H	Y	B	D
E	F	G	I , J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair Cipher (2/3)

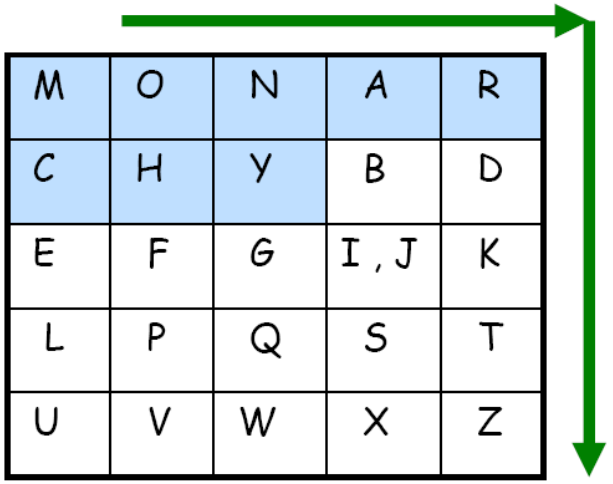
- Encrypts two letters at a time
  - If a pair is a repeated letter, insert a filler like "x".

    E.g. "balloon" encrypts as "ba lx lo on"
  - If both letters fall in the same row, replace each with letter to right (wrapping back to start from end).

    E.g. "ar" encrypts as "RM"
  - If both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom).

    E.g. "mu" encrypts to "CM"
  - Otherwise each letter is replaced by the one in its row in the column of the other letter of the pair.

    E.g. "hs" encrypts to "BP"



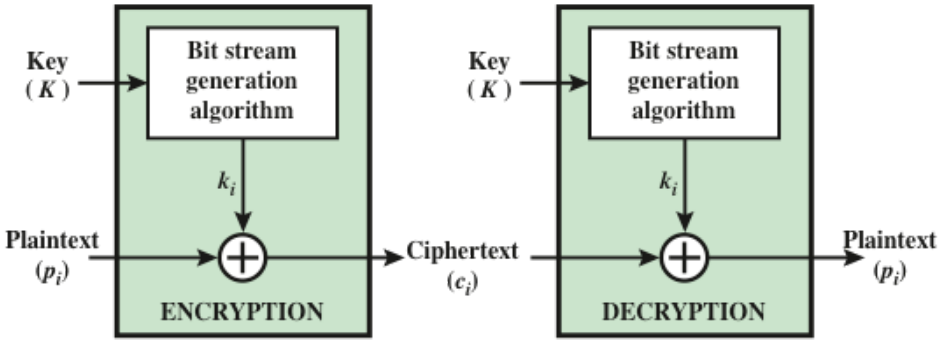
Playfair Cipher (3/3)

- Security is greatly improved over substitution ciphers.
  - Since have 26 x 26 = 676 digrams
  - It needs a 676 entry frequency table to analyze (verses 26 for a substitution cipher)

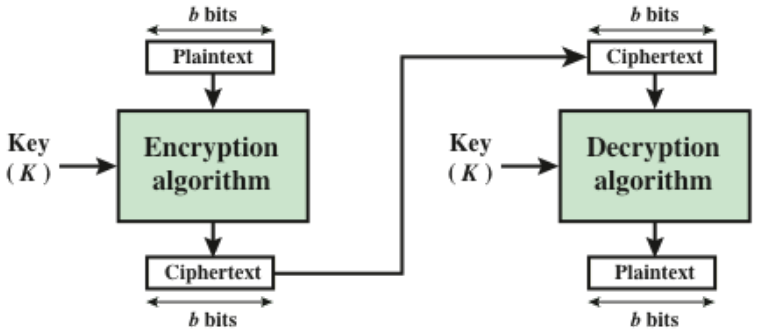
- Playfair Ciper was widely used for many years (eg. US & British military in World War One)

- It CAN be broken, given a few hundred letters
  - Since it still has much of plaintext structure.

Stream Cipher and Block Cipher



(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

Figure 3.1 Stream Cipher and Block Cipher

Block Ciphers

- In general, a block cipher replaces a block of N plaintext bits with a block of N ciphertext bits. (E.g., N = 64 or 128.)
- A block cipher is a substitution cipher.
- Each block may be viewed as a gigantic character.
- The "alphabet" consists of 2^N gigantic characters.
- Each particular cipher is a one-to-one mapping from the plaintext "alphabet" to the ciphertext "alphabet".
- There are $2^N!$ such mappings.
- A secret key indicates which mapping to use.

Ideal Block Cipher

- An ideal block cipher would allow us to use any of these $2^N!$ mappings.
  - The key space would be extremely large.
- But this would require a key of $\log_2(2^N!)$ bits.
- If N = 64,

  $\log_2(2^N!) \approx N \times 2^N \approx 10^{21}$ bits $\approx 10^{11}$ GB.
- Infeasible!

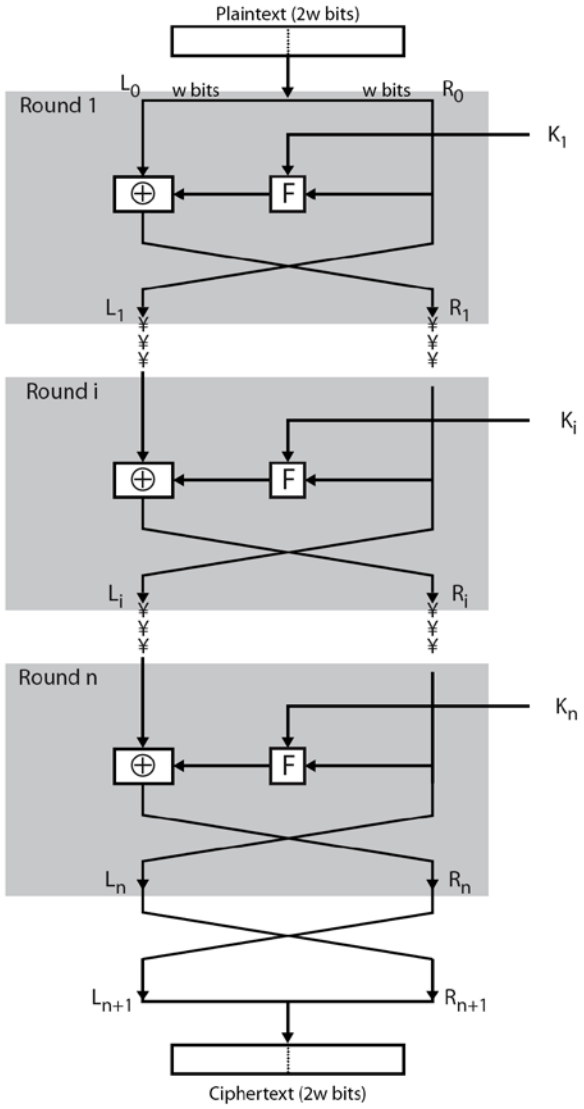
Practical Block Ciphers

- Modern block ciphers use a key of K bits to specify a random subset of 2^K mappings.
- If $K \approx N$,
  - 2^K is much smaller than 2^N!
  - But is still very large.
- If the selection of the 2^K mappings is random, the resulting cipher will be a good approximation of the ideal block cipher.
- Horst Feistel, in1970s, proposed a method to achieve this.

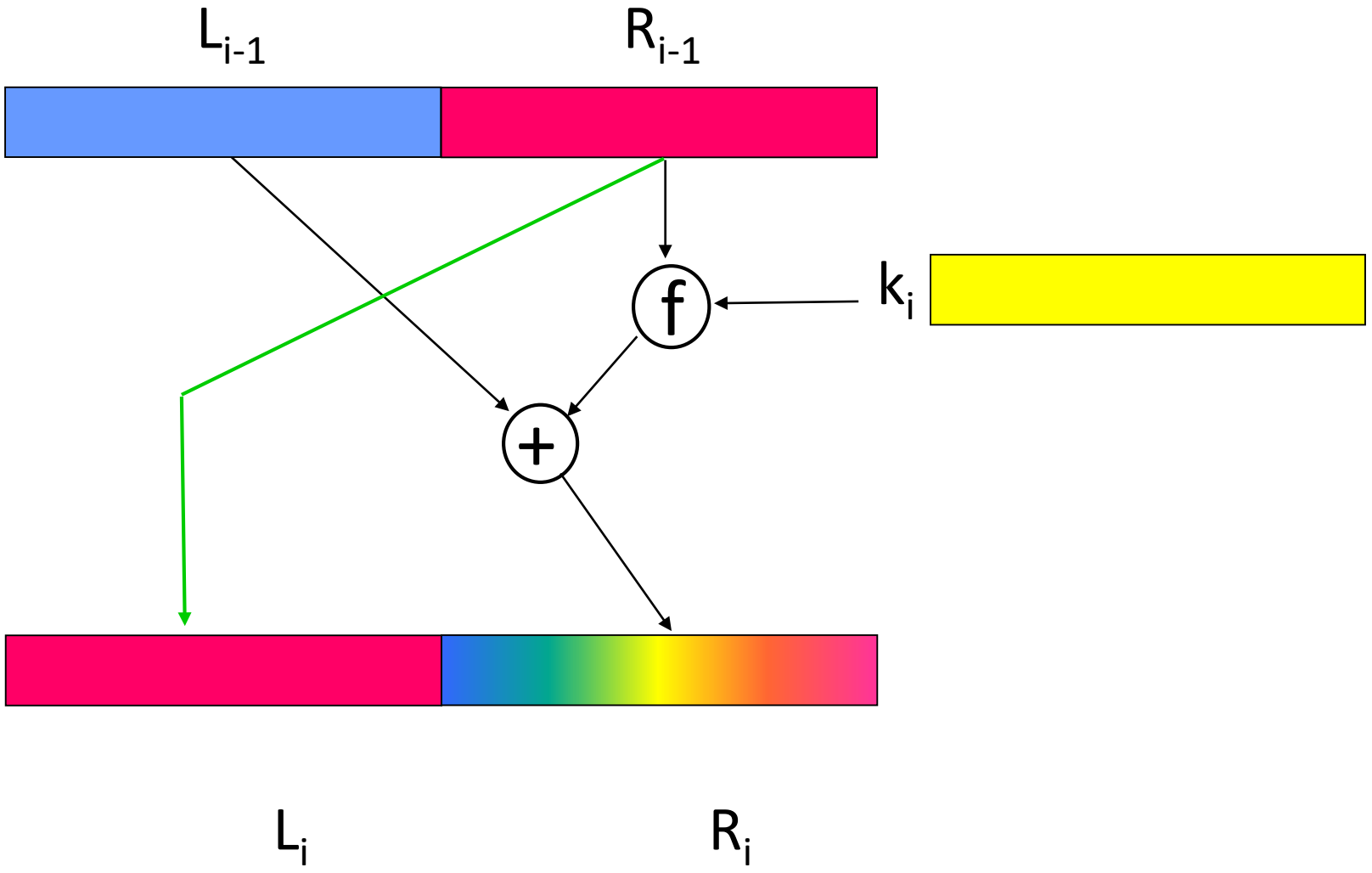
The Feistel Cipher Structure

- Input:
  - a data block and
  - a key
- Partition the data block into two halves L and R.
- Go through a number of rounds.
- In each round,
  - R does not change.
  - L goes through an operation that depends on R and a round key derived from the key.

The Feistel Cipher Structure



Round i



Mathematical Description of Round i

- Let L_{i-1} and R_{i-1} be the input of round i, and

  L_i and R_i the output.

- We have

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

- Or, $(L_i, R_i) := \mu \circ \phi_i (L_{i-1}, R_{i-1})$, where

$$\phi_i : (x, y) \rightarrow (x \oplus F(y, k_i), y).$$
$$\mu : (x, y) \rightarrow (y, x).$$

- Note that $\phi_i^{-1} = \phi_i$ and $\mu^{-1} = \mu$.

$$x \oplus F(y, k_i) \oplus F(y, k_i) = x$$

Feistel Cipher

- Goes through a number of rounds, say 16 rounds.
- A Feistel cipher encrypts a plaintext block m as:

$$c := \mathrm{E}_k(m) := \mu \circ \mu \circ \phi_{16} \circ \cdots \circ \mu \circ \phi_2 \circ \mu \circ \phi_1(m)$$

- The decryption will be:

$$\mathrm{D}_k(c) = \phi_1^{-1} \circ \mu^{-1} \circ \phi_2^{-1} \circ \cdots \circ \mu^{-1} \circ \phi_{16}^{-1} \circ \mu^{-1} \circ \mu^{-1}(c)$$

$$= \mu \circ \mu \circ \phi_1 \circ \mu \circ \phi_2 \circ \cdots \circ \mu \circ \phi_{16}(c)$$

- The descryption algorithm is the same as the encryption algorithm, but uses round keys in the reverse order.

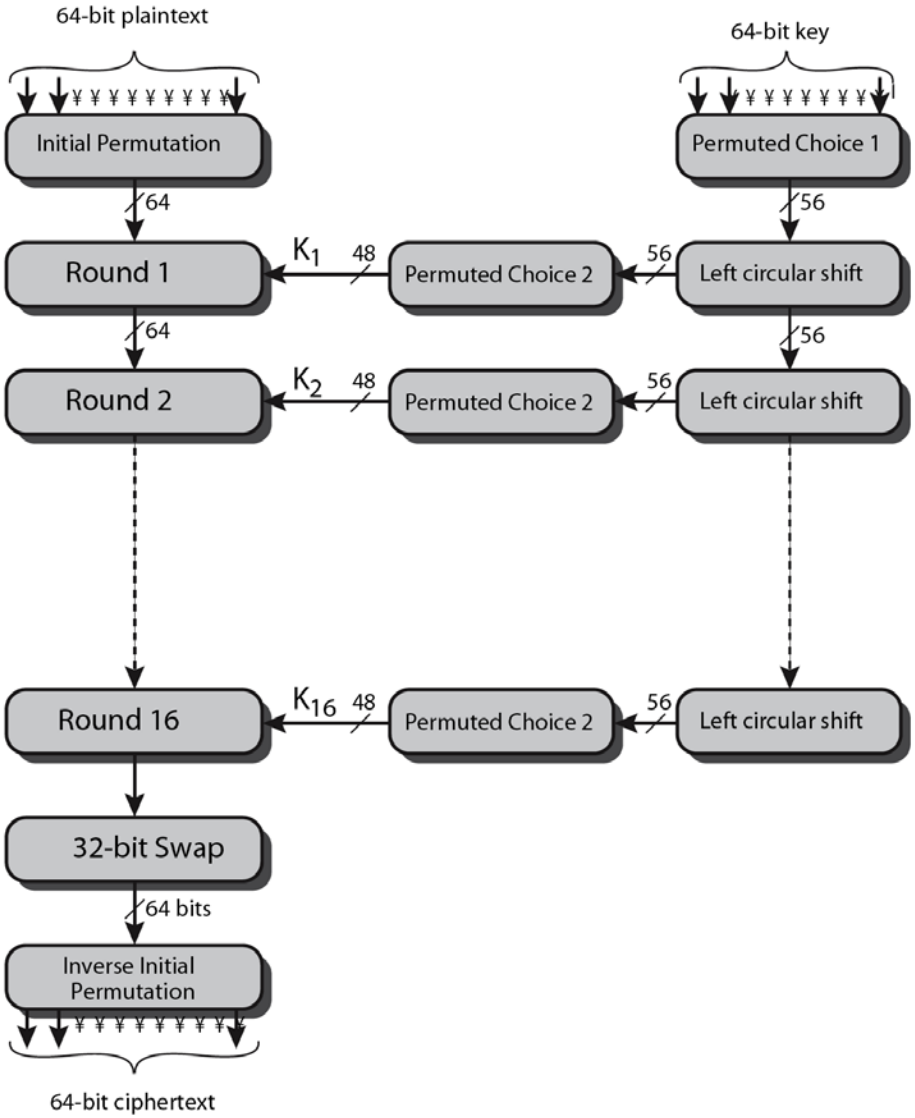
DES: The Data Encryption Standard

- Most widely used block cipher in the world.
- Adopted by NIST in 1977.
- Based on the Feistel cipher structure with 16 rounds of processing.
- Block = 64 bits
- Key = 56 bits
- What is specific to DES is *the design of the F function* and *how round keys are derived from the main key*.

Design Principles of DES

- To achieve high degree of diffusion and confusion.
- **Diffusion**: making each plaintext bit affect as many ciphertext bits as possible.
- **Confusion**: making the relationship between the encryption key and the ciphertext as complex as possible.

DES Encryption Overview



Round Keys Generation (1/3)

- Main key: 56 bits.
  - 56 = 7 bits * 8
  - Append one odd parity bit to each 7 bits.
  - Now, the key looks as (7+1) * 8 = 64 bits

Round Keys Generation (2/3)

- 56-bits are permuted using Permuted Choice One (PC1), and then divided into two 28-bit halves.

C_0

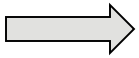
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36

D_0

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

- The permutation is not random

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



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

- The permutation has no security value.

Round Keys Generation (3/3)

- In each round:
  - Left-rotate **each half** separately by either 1 or 2 bits according to a rotation schedule.
    - In rounds 1, 2, 9, 16: left-rotate 1 bit.
    - In other rounds: left-rotate 2 bits.
  - Select 24-bits from each half, and permute the combined 48 bits.
    - The permutation of C_i is the following: (note that bits 9, 18, 22, and 25 are discarded)

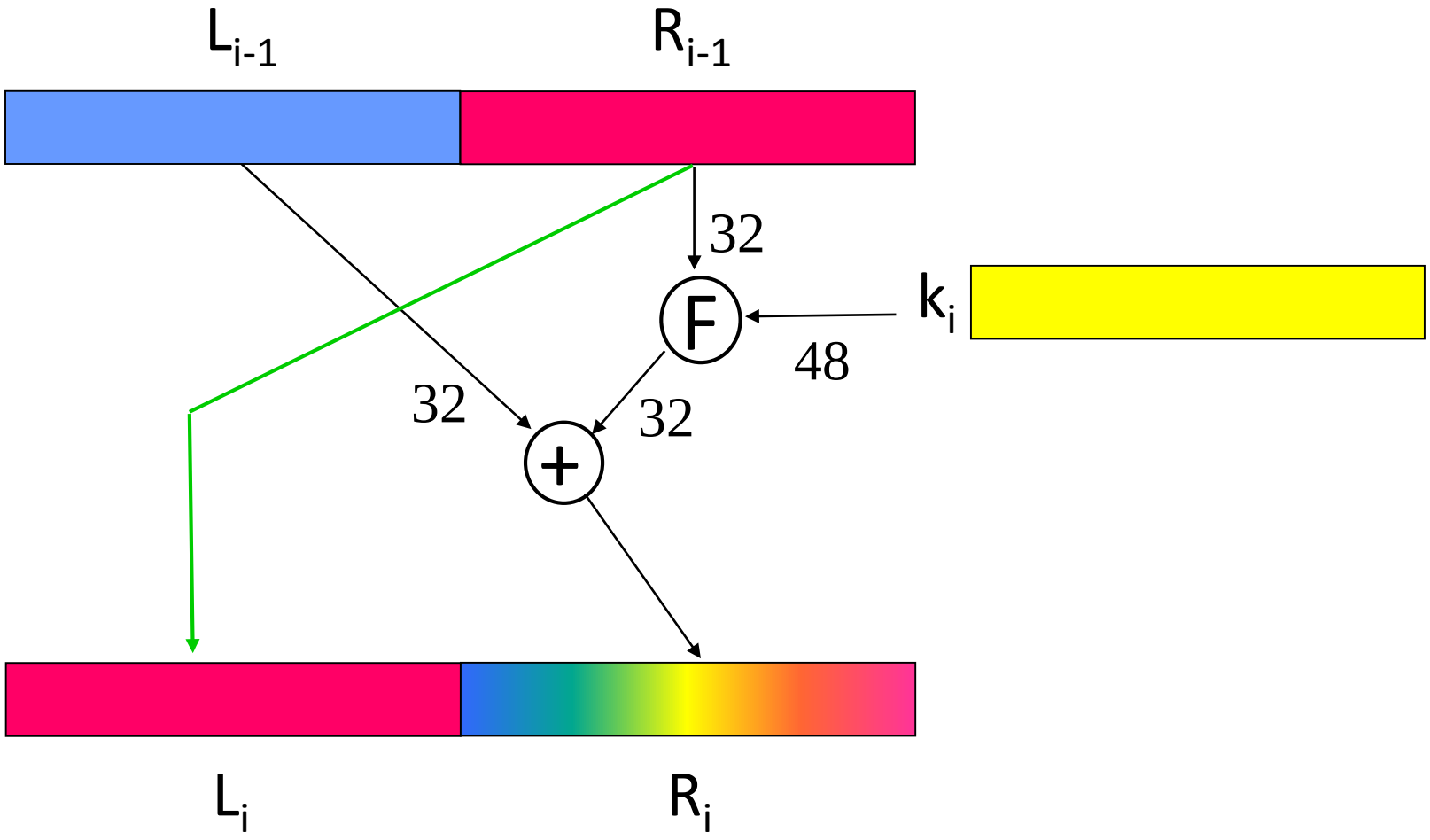
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2

    - The permutation of D_i is the following: (note that bits 35, 38, 43, and 54 are discarded)

41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

  - This forms a round key: $K_i = C_i \mid D_i$

Round i



The F function of DES

- The L and R each have 32 bits, and the round key K 48 bits.
- The F function, on input R and K, produces 32 bits:

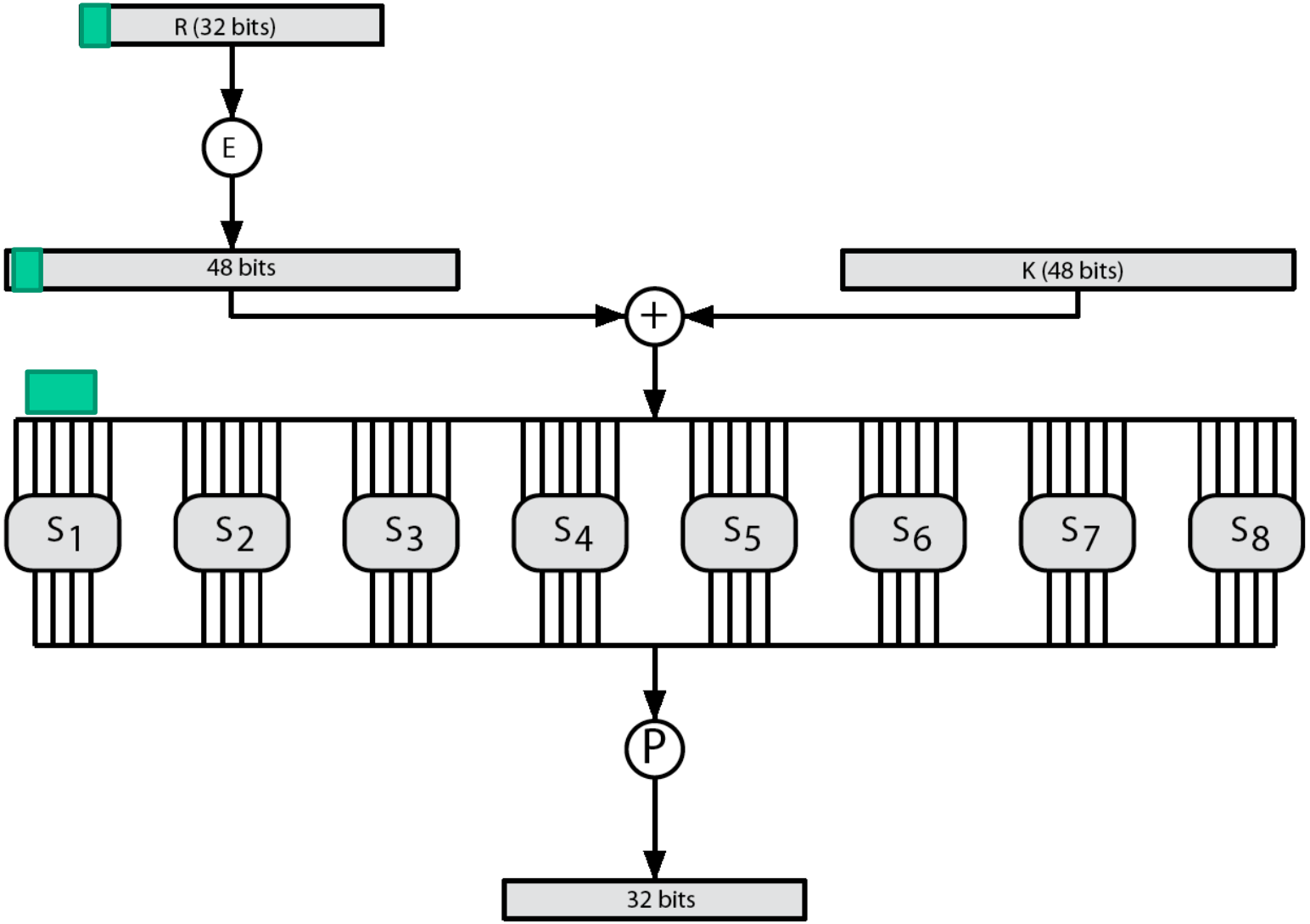
$$F(R, K) = P\big(S\big(E(R) \oplus K\big)\big)$$

where E: expands 32 bits to 48 bits;

S: shrinks it back to 32 bits;

P: permutes the 32 bits.

The F function of DES



The 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

The S-Boxes

- Eight S-boxes each map 6 to 4 bits
- Each S-box is specified as a 4 x 16 table
  - each row is a permutation of 0-15
  - outer bits 1 & 6 of input are used to select one of the four rows
  - inner 4 bits of input are used to select a column
- All the eight boxes are different.
- How these boxes are designed?
  - Secret!

Example: S-box 1

	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
1	0	15	7	4	14	2	13	1	10	6	12	11	6	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

- Example S-box lookup:
  - $S_1(101010) = 6$

Permutation Function P

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

Initial Permutation IP

- IP: the first step of the encryption.
- It reorders the input data bits.
- The last step of encryption is the inverse of IP.
- IP and IP^{-1} are specified by tables

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

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

IP and IP^{-1} have no security value

- Let's call a modified DES without these two permutations **A**.
- We prove that **A** and DES have the same security as follows.
  - Suppose these permutation are important, which means that we can break **A** but cannot break DES.
  - Let's say we can break **A**, i.e., given a plaintext-ciphertext pair <m, c>, we can calculate the **A** key k.
  - In this case, we can easily break DES as well
  - Given a DES pair <m, c>, we have $c=IP^{-1}(\mathbf{A}(IP(m)))$, which means that $IP(c)=\mathbf{A}(IP(m))$. So, given <IP(m), IP(c)>, we can calculate the **A** key k, which is also DES key k.

DES Example

Round	*Ki*	*Li*	*Ri*
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP–1		da02ce3a	89ecac3b

Note: DES subkeys are shown as eight 6-bit values in hex format

Avalanche Effect

- Avalanche effect:
  - A small change in the plaintext or in the key results in a significant change in the ciphertext.
  - an evidence of high degree of diffusion and confusion
  - a desirable property of any encryption algorithm
- DES exhibits a strong avalanche effect
  - Changing 1 bit in the plaintext affects 34 bits in the ciphertext on average.
  - 1-bit change in the key affects 35 bits in the ciphertext on average.

Avalanche Effect in DES: Change in Plaintext

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

Avalanche Effect in DES: Change in Key

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30

Block Cipher Design Principles: Number of Rounds

The greater the number of rounds, the more difficult it is to perform cryptanalysis

In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack

If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search

Block Cipher Design Principles: Design of Function F

- The heart of a Feistel block cipher is the function F
- The more nonlinear F, the more difficult any type of cryptanalysis will be
- The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function

- The algorithm should have good avalanche properties

Strict Avalanche Criterion (SAC)

States that any output bit j of an S-box should change with probability 1/2 when any single input bit i is inverted for all i , j

Bit Independence Criterion (BIC)

States that output bits j and k should change independently when any single input bit i is inverted for all i , j , and k

Block Cipher Design Principles: Key Schedule Algorithm

- With any Feistel block cipher, the key is used to generate one subkey for each round
- In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key
- It is suggested that, at a minimum, the key schedule should guarantee key/ciphertext **Strict Avalanche Criterion** and **Bit Independence Criterion**

Attacks on DES

- Brute-force key search
  - Trying 1 key per microsecond would take 1000+ years on average, due to the large key space size, $2^{56} \approx 7.2 \times 10^{16}$.
- Differential cryptanalysis
  - Possible to find a key with 2^{47} plaintext-ciphertext samples
  - Known-plaintext attack
- Liner cryptanalysis:
  - Possible to find a key with 2^{43} plaintext-ciphertext samples
  - Known-plaintext attack

DES Cracker

- DES Cracker:
  - A DES key search machine
  - contains 1536 chips
  - Cost: $250,000.
  - could search 88 billion keys per second
  - won RSA Laboratory's "DES Challenge II-2" by successfully finding a DES key in 56 hours.
- DES is feeling its age. A more secure cipher is needed.

Multiple Encryption with DES

- In 2001, NIST published the Advanced Encryption Standard (AES) to replace DES.
- But users in commerce and finance are not ready to give up on DES.
- As a temporary solution to DES's security problem, one may encrypt a message (with DES) multiple times using multiple keys:
  - 2DES is not much securer than the regular DES
  - So, 3DES with either 2 or 3 keys is used

2DES

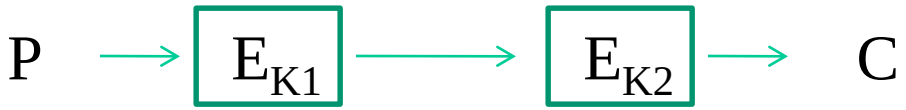
- Consider 2DES with two keys:

  $C = E_{K2}(E_{K1}(P))$

- Decryption: $P = D_{K1}(D_{K2}(C))$
- Key length: 56 x 2 = 112 bits
- This should have thwarted brute-force attacks?
- Wrong!

Meet-in-the-Middle Attack on 2DES

- 2-DES: $C = E_{K2}(E_{K1}(P))$



- Given a known pair (P, C), attack as follows:
  - Encrypt P with all 2^{56} possible keys for K1.
  - Decrypt C with all 2^{56} possible keys for K2.
  - If $E_{K1'}(P) = D_{K2'}(C)$, try the keys on another (P', C').
  - If works, (K1', K2') = (K1, K2) with high probability.
  - Takes $O(2^{56})$ steps; not much more than attacking 1-DES.

3 DES with 2 Keys

- A straightforward implementation would be:

$$c = E_{k_3}(E_{k_2}(E_{k_1}(m)))$$

- In practice: $c = E_{k_3}(D_{k_2}(E_{k_1}(m)))$
- Reason: for backward compatibility: 3DES software can be used as 1DES software or 3DES with 2 keys
  - If $k_1 = k_2$, then becomes 1DES.
  - If $k_1 = k_3$, then becomes 3DES with 2 keys.
- Standardized in ANSI X9.17 & ISO 8732
- No practical attacks are known.
- Some Internet applications, such as PGP, uses 3DES with 3 keys.

Homework

- Textbook Chapter 3, problems:
  - 3.3 and 3.6.