



# Systems and Internet Infrastructure Security

Network and Security Research Center  
Department of Computer Science and Engineering  
Pennsylvania State University, University Park PA

## An introduction to Block Ciphers

Presentation by – Wesam Lootah  
Oct 2005

# Symmetric-Key Block Ciphers

- When used “properly” can protect the secrecy of data and communication
- Building block in many secure protocols
  - IPsec, SSH, SSL
- Also plays a role in authentication protocols
  - Kerberos
- Used as pseudo random numbers generators
- Considerably faster than public-key ciphers



# Block Ciphers

- Block ciphers operate on a **block** of input and produce a block of output.
- They can be viewed as a simple substitution cipher with large character size.
- The most **general** block cipher implements every possible substitution. For a  $n$ -bit block cipher there are  $(2^n!)$  Substitutions (bijections)
- The key of such a general cipher requires  $\sim \lg(2^n!)$  bits

# Block Ciphers

- Modern block ciphers do not implement all  $2^n!$  Substitutions.
- Most use  $n$ -bit keys and implement  $2^n$  substitutions.
- Most modern block ciphers are **iterated product ciphers**
- Why iterated? Why product?



# Product Ciphers

- Endomorphic cipher:  
 a cipher where  $P = C$
  
- Ciphers can be combined to form product ciphers
  - ▶  $S_1 = \{P, P, K_1, E_1, D_1\}$  and  $S_2 = \{P, P, K_2, E_2, D_2\}$
  - ▶ The product of  $S_1$  and  $S_2$ , denoted by  $S_1 \times S_2$
  - ▶  $S_1 \times S_2 = \{P, P, K_1 \times K_2, E, D\}$  where
  - ▶  $e_{(K_1, K_2)} = e_{K_2}(e_{K_1}(x))$
  - ▶  $d_{(K_1, K_2)} = d_{K_1}(d_{K_2}(x))$
  
- A product of  $S$  with itself is denoted  $S \times S$  or  $S^2$

# Idempotent Cryptosystem

- If  $S$  is a cryptosystem and  $S^2 = S$ , then  $S$  is idempotent
- Example: permutations are idempotent
  - A permutation applied twice can be represented by a single permutation
  - For example the permutation (2 4 1 5 6 3) applied twice is (4 5 2 6 3 1)
- Non-Idempotent cryptosystems may be constructed by taking the product of two different (simple) cryptosystems.

# Block Cipher: Encryption

- Encryption is done in  $N_r$  rounds
- The Key is used to generate  $N_r$  round keys.
  - $(K^1 \dots K^{N_r})$  is called the *key schedule*
  - $K^i$  is called a round key or a sub-key
- The cipher uses a *round* function  $g$ 
  - $g$  takes two inputs: a sub-key and the output of the previous round.
  - The input to the first round is the plain text and  $K^1$  and the output of the last round is the cipher text

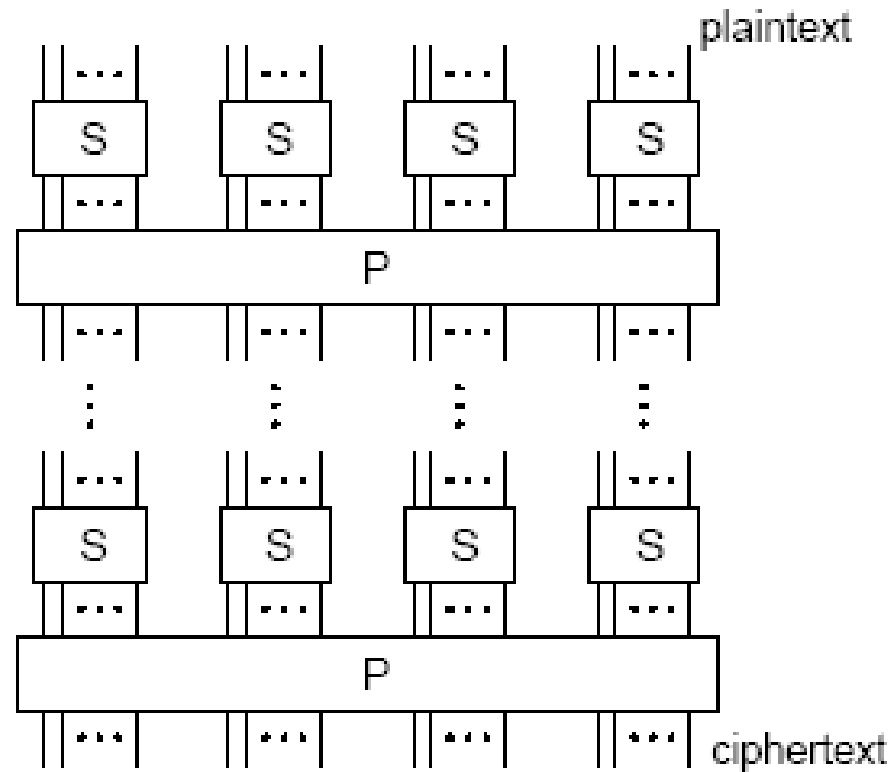
# Block Cipher: Decryption

- Decryption is also done in  $N_r$  rounds
- The key schedule is used in reverse order
- Decryption uses the inverse of  $g$ , the round function
  - $g$  has to be **injective** (one-to-one) to be invertible
- Input to the first round is the cipher text and  $K^{N_r}$
- The output of the last round is the plain text

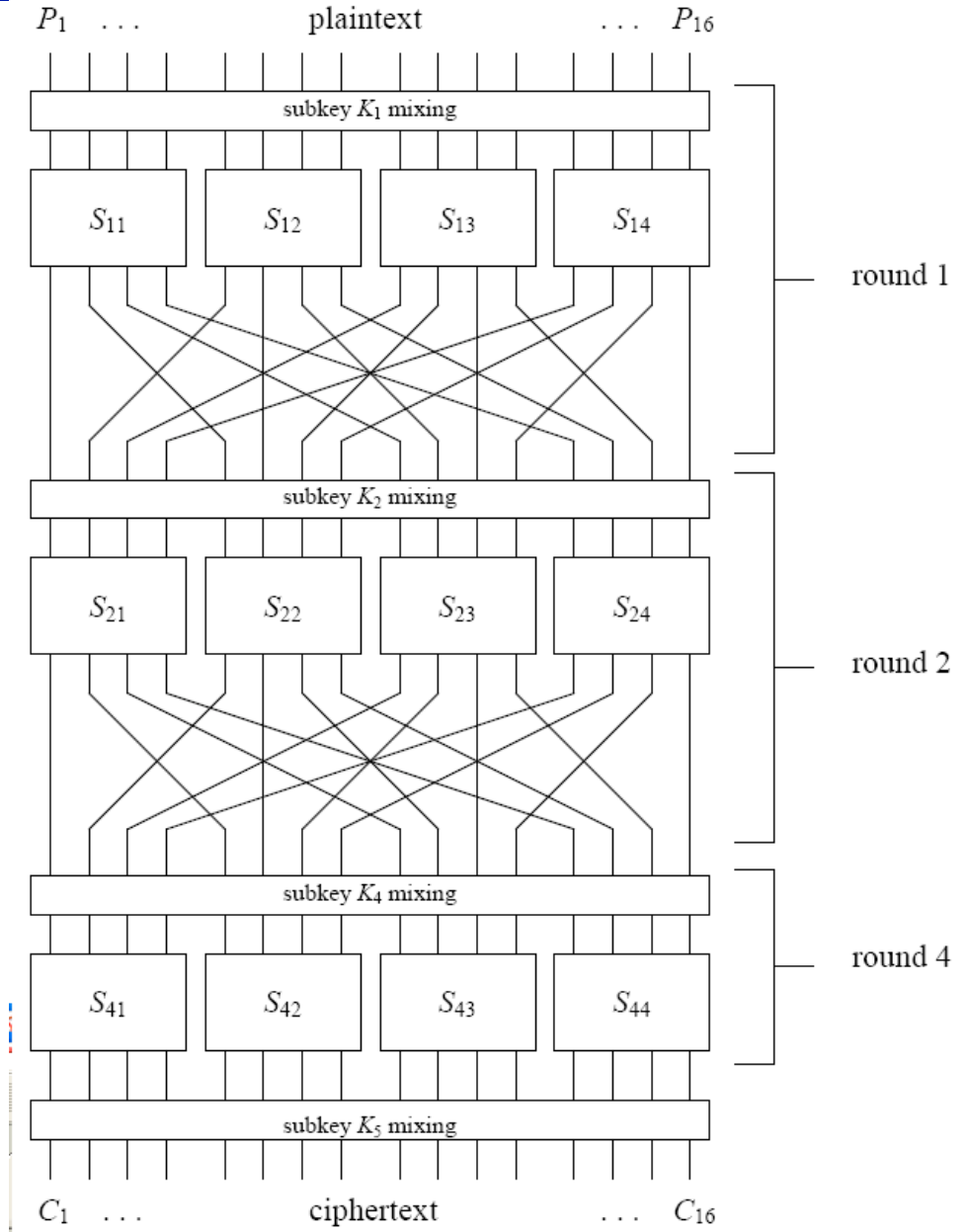
- Substitution Permutation Network (SPN) Ciphers
- Feistel Ciphers

# Substitution-Permutation Networks (SPN)

- Is a product cipher composed of a number of stages each involving substitution and permutation



# Sample SPN



# A Simple SPN

For example: a 16-bit SPN with 16-bit key that uses the following:

## Substitution

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

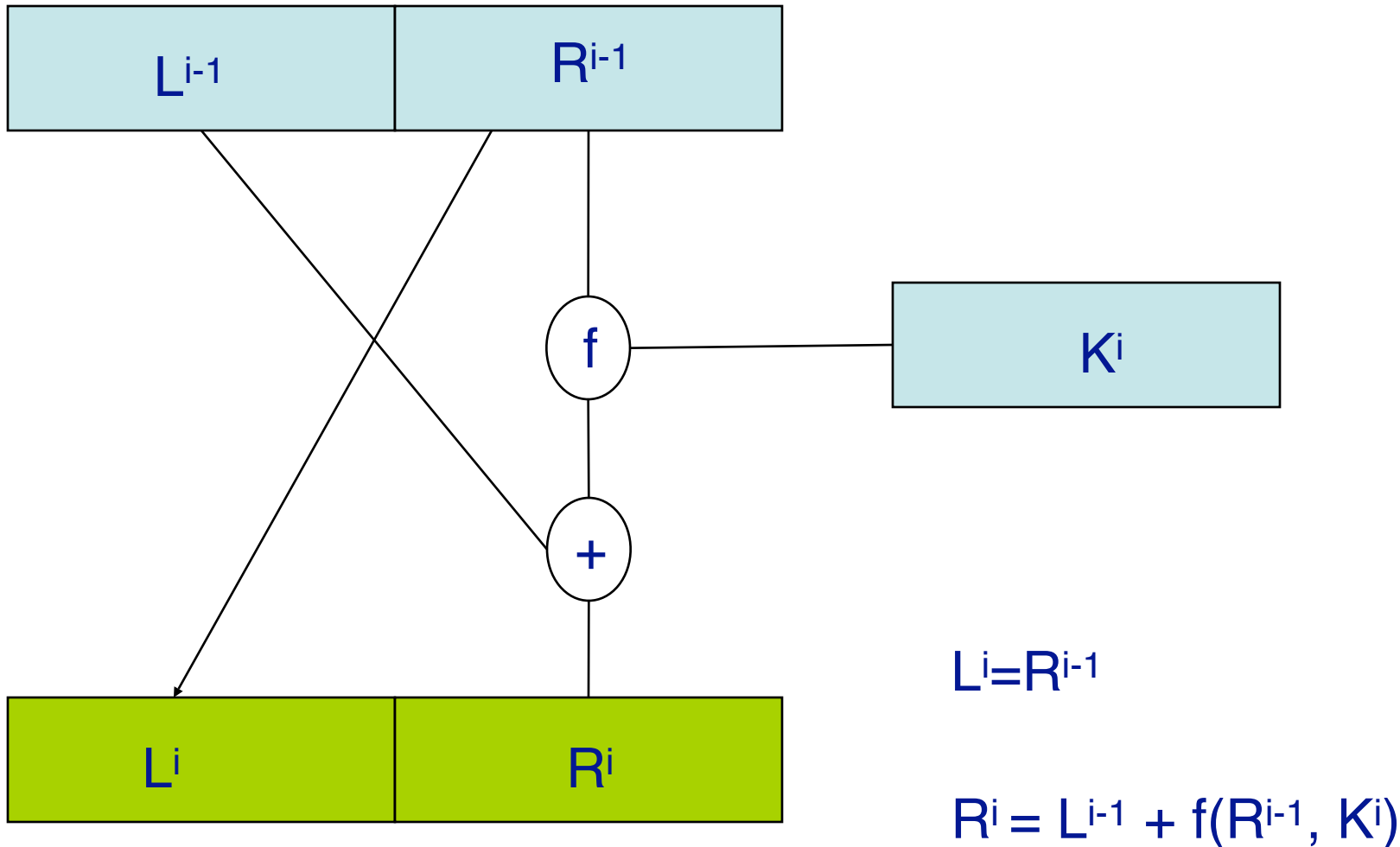
## Permutation

	1	2	3	4	5	6	7	8	9	1	11	1	1	14	1	1
										0		2	3		5	6
	1	5	9	13	2	6	1	1	3	7	11	1	4	8	1	1
							0	4				5			2	6

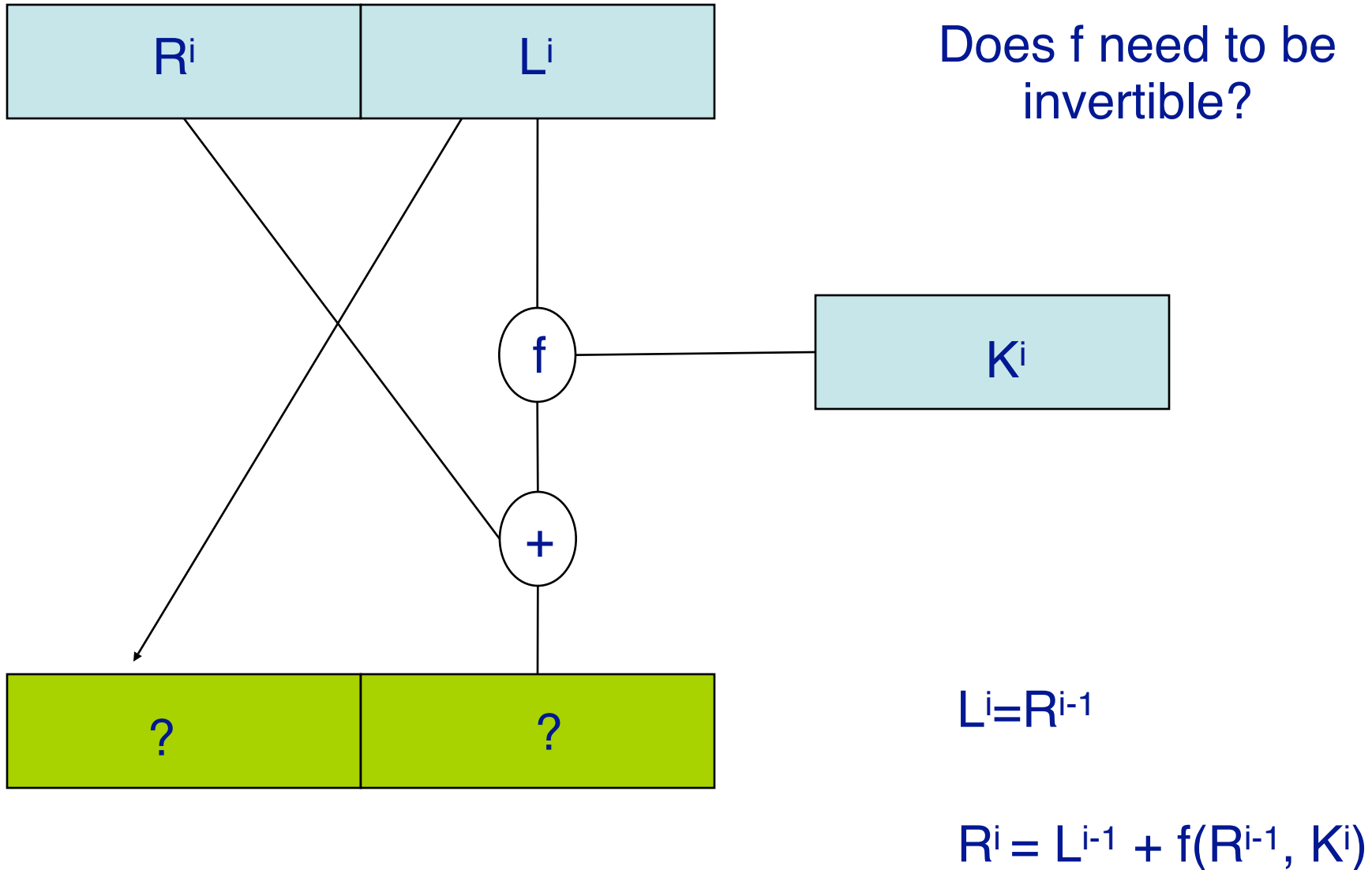
Can we use substitutions that are Not bijections?

# Feistel cipher

## Round function in a Feistel cipher



# Feistel cipher



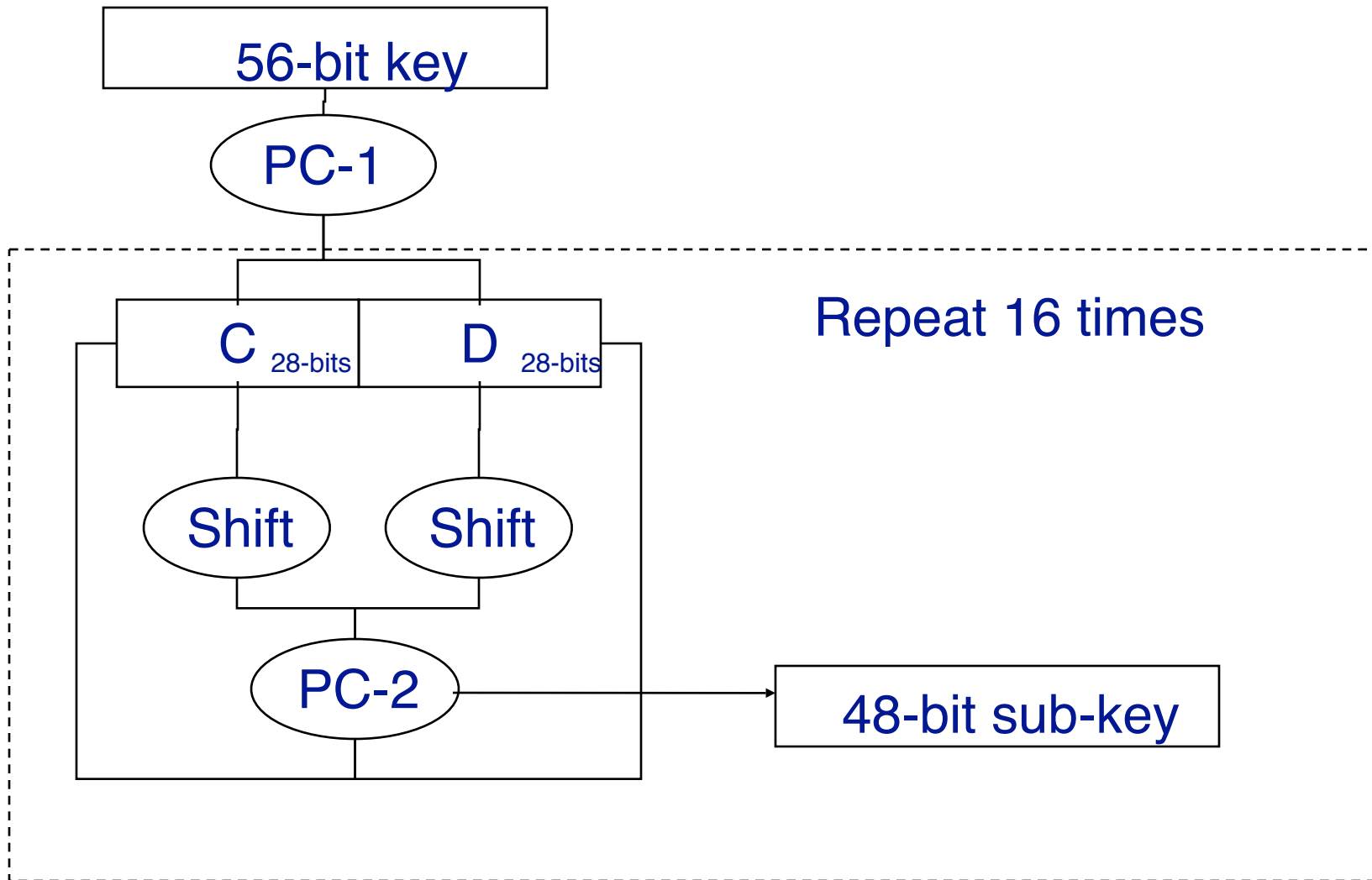
# Data Encryption Standard

- DES is a Feistel cipher
- It was the most widely used cryptosystem in the world
- DES operates on 64-bit blocks and uses a 56-bit keys
- Electronic Frontier Foundation created a \$220,000 machine to crack DES-encrypted messages

# DES Key schedule

- Key is represented by 64-bits, however only 56 are used.
- Key schedule:
  - Permute the key according to PC-1 (result is 56 bits)
  - Split the key into two halves L and R
  - Loop (16 times)
    - Shift L and R left 2 bits (Except for 1,2, 9 and 16 shift left once)
    - Now use L and R to create sub-key by permuting according to PC-2 (which only uses 48 of the 56 bits)
- Each sub-key is 48-bits long

# DES: Key Schedule



# DES: Round

- Permute 64-bit block according to IP
- Split into R and L (each 32 bits long)
- Apply Feistel network
- F is:
  - Expand 32 bits to 48 bits
  - XOR with 48-bit sub-key (key mixing)
  - 48 bits are split into 8 groups of 6
  - Each 6 bits is input to an S-Box which outputs only 4 bits
  - The result 32-bits is permuted (P)

# DES: Round

- Repeat in 16 rounds
- At the end of 16<sup>th</sup> round reverse L and R
- And permute using  $IP^{-1}$  which is the inverse of IP

# Linear Cryptanalysis

- Linear cryptanalysis takes advantage of high probability occurrences of *linear* expressions involving plaintext bits, ciphertext bits and subkey bits.
- It is a *known plaintext* attack. We assume the attacker has access to a large number of plaintext and ciphertext pairs.
- We approximate a portion of the cipher with a linear expressio.

- Linear expressions refer to expressions of the form:

$$X_{i_1} + X_{i_2} + \dots + X_{i_u} + Y_{j_1} + Y_{j_2} + \dots + Y_{j_v} = 0$$

Where + denotes bitwise exclusive-OR

X is the *i*th input bit and Y is the *j*th output bit

Linear cryptanalysis depends on finding expressions of the form above that occur with a high or low probability

The bias of a linear expression is the amount its probability deviates from  $\frac{1}{2}$

# Piling-Up Principle

- Consider two random binary variables  $X_1$  and  $X_2$
- $X_1 + X_2 = 0$  means  $X_1 = X_2$
- Assume probability distribution are given by:

$$\Pr(X_1 = i) = \begin{cases} p_1 & , i = 0 \\ 1 - p_1 & , i = 1 \end{cases}$$

$$\Pr(X_2 = i) = \begin{cases} p_2 & , i = 0 \\ 1 - p_2 & , i = 1. \end{cases}$$

# Piling-Up

- If the two random variables are independent, then

$$\Pr(X_1 = i, X_2 = j) = \begin{cases} p_1 p_2 & , i = 0, j = 0 \\ p_1(1 - p_2) & , i = 0, j = 1 \\ (1 - p_1)p_2 & , i = 1, j = 0 \\ (1 - p_1)(1 - p_2) & , i = 1, j = 1 \end{cases}$$

- and it can be shown that

$$\begin{aligned} \Pr(X_1 \oplus X_2 = 0) &= \Pr(X_1 = X_2) \\ &= \Pr(X_1 = 0, X_2 = 0) + \Pr(X_1 = 1, X_2 = 1) \\ &= p_1 p_2 + (1 - p_1)(1 - p_2). \end{aligned}$$

# Pilling-Up

- We let :

$$p_1 = \frac{1}{2} + \varepsilon_1$$

$$p_2 = \frac{1}{2} + \varepsilon_2$$

where  $\varepsilon_1$  and  $\varepsilon_2$  are the probability biases

$$-\frac{1}{2} \leq \varepsilon_1 \leq \frac{1}{2} \quad \varepsilon_2 \leq \frac{1}{2}$$

$$\Pr(X_1 \oplus X_2 = 0) = \frac{1}{2} + 2\varepsilon_1\varepsilon_2$$

The bias  $\varepsilon_{1,2}$  of  $X_1 + X_2 = 0$  is

$$\varepsilon_{1,2} = 2\varepsilon_1\varepsilon_2.$$

# Piling-Up

## Piling-Up Lemma (Matsui [1])

For  $n$  independent, random binary variables,  $X_1, X_2, \dots, X_n$ ,

$$\Pr(X_1 \oplus \dots \oplus X_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^n \epsilon_i$$

or, equivalently,

$$\epsilon_{1,2,\dots,n} = 2^{n-1} \prod_{i=1}^n \epsilon_i$$

where  $\epsilon_{1,2,\dots,n}$  represents the bias of  $X_1 \oplus \dots \oplus X_n = 0$ .

- By combining representations of S-Boxes we can eliminate some intermediate steps.

# Linear Approximations

		Output Sum															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Input Sum	0	+8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	-2	-2	0	0	-2	+6	+2	+2	0	0	+2	+2	0	0
	2	0	0	-2	-2	0	0	-2	-2	0	0	+2	+2	0	0	-6	+2
	3	0	0	0	0	0	0	0	0	+2	-6	-2	-2	+2	+2	-2	-2
	4	0	+2	0	-2	-2	-4	-2	0	0	-2	0	+2	+2	-4	+2	0
	5	0	-2	-2	0	-2	0	+4	+2	-2	0	-4	+2	0	-2	-2	0
	6	0	+2	-2	+4	+2	0	0	+2	0	-2	+2	+4	-2	0	0	-2
	7	0	-2	0	+2	+2	-4	+2	0	-2	0	+2	0	+4	+2	0	+2
	8	0	0	0	0	0	0	0	0	-2	+2	+2	-2	+2	-2	-2	-6
	9	0	0	-2	-2	0	0	-2	-2	-4	0	-2	+2	0	+4	+2	-2
	A	0	+4	-2	+2	-4	0	+2	-2	+2	+2	0	0	+2	+2	0	0
	B	0	+4	0	-4	+4	0	+4	0	0	0	0	0	0	0	0	0
	C	0	-2	+4	-2	-2	0	+2	0	+2	0	+2	+4	0	+2	0	-2
	D	0	+2	+2	0	-2	+4	0	+2	-4	-2	+2	0	+2	0	0	+2
	E	0	+2	+2	0	-2	-4	0	+2	-2	0	0	-2	-4	+2	-2	0
	F	0	-2	-4	-2	-2	0	+2	0	0	-2	+4	-2	-2	0	+2	0

Table 4. Linear Approximation Table

- Question?