

Encryption Algorithms

- 1. Introduction**
- 2. Cryptographic Systems**
- 3. Security of a Cryptosystem**
- 4. Transposition Ciphers**
- 5. Substitution Ciphers**
- 6. Product Ciphers**
- 7. Exponentiation Ciphers**
- 8. Cryptography based on Discrete Logarithms**
- 9. Advanced Encryption Standard (AES)**
- 10. Cryptographic Techniques**

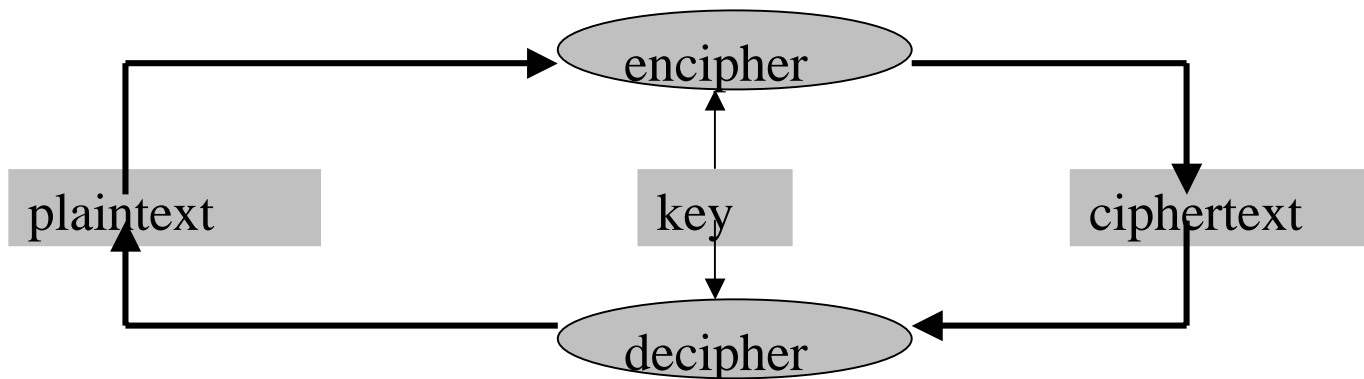
1. Introduction

Definitions

Cryptology = Cryptography + Cryptanalysis

Cryptography: science and study of secret writing.

Cipher: a secret method of writing, whereby *plaintext* (or cleartext) is transformed into *ciphertext*.



-There are two basic types of ciphers: *transpositions* and *substitutions*.

- *Transposition ciphers* rearrange bits of characters in the data.
- *Substitution ciphers* replace bits, characters, or blocks of characters with substitutes.

Example of Substitution Ciphers: Caesar cipher

-Shifts each letter in the English alphabet forward by K positions;
K is the key to the cipher (e.g. K=3).

IMPATIENT WAITER



LPSDWLHQW ZDLWHU

Example of Transposition Ciphers: Rail-fence

- The letters of a plaintext message are written down in a pattern resembling a rail fence, and then removed by rows.
- The key to the cipher is given by the depth of the fence (e.g =3)

DISCONCERTED COMPOSER



D O R C O
 I C N E T D O P S R
 S C E M E



DORCOICNETDOPSRSCEME

Cryptanalysis: science and study of methods of breaking ciphers. A cipher is **breakable** if it is possible to determine the plaintext from the ciphertext or the key from the plaintext-ciphertext pairs.

-Four basic methods of attacks: *ciphertext-only*, *known-plaintext*, *chosen-plaintext*, and *chosen-ciphertext*.

- *Ciphertext-only attack*: a cryptanalyst must determine the key solely from intercepted ciphertext.
- *Known-plaintext attack*: a cryptanalyst knows some plaintext-ciphertext.
- *Chosen-plaintext attack*: a cryptanalyst is able to acquire the ciphertext corresponding to selected plaintext.
- *Chosen-ciphertext attack*: applicable only to public key systems; the cryptanalyst uses the plaintext to deduce the key.

Example: Ciphertext-only Attack on Substitution Ciphers

- Table of English character frequencies by D. Denning

a.	0.080	h.	0.060	n.	0.070	t.	0.090
b.	0.015	i.	0.065	o.	0.080	u.	0.030
c.	0.030	j.	0.005	p.	0.020	v.	0.010
d.	0.040	k.	0.005	q.	0.002	w.	0.015
e.	0.130	l.	0.035	r.	0.065	x.	0.005
f.	0.020	m.	0.030	s.	0.060	y.	0.020
g.	0.015					z.	0.002

-The correlation of the frequency of each letter i in the ciphertext with the character frequencies in English denoted $\phi(i)$ is obtained as:

$$\phi(i) = \sum_{c=0}^{25} f(c) p(c-i)$$

-Example: Cryptanalyse the cipher “KHOOR ZRUOG” assuming that it as obtained using a Caesar cipher.

•Frequency analysis $f(c)$:

G 0.1 H 0.1 K 0.1 O 0.3 R 0.2 U 0.2 Z 0.1

•Correlation:

$$\begin{aligned}\phi(i) &= \sum_{c=1}^{25} f(c)p(c-i) \\ &= 0.1p(6-i) + 0.1p(7-i) + 0.1p(10-i) + 0.3p(14-i) \\ &\quad + 0.2p(17-i) + 0.1p(20-i) + 0.1p(25-i)\end{aligned}$$

a.	0.0482	h.	0.0442	n.	0.0520	t.	0.0315
b.	0.0364	i.	0.0202	o.	0.0535	u.	0.0302
c.	0.0410	j.	0.0267	p.	0.0226	v.	0.0517
d.	0.0575	k.	0.0635	q.	0.0322	w.	0.0380
e.	0.0252	l.	0.0262	r.	0.0392	x.	0.0370
f.	0.0190	m.	0.0325	s.	0.0299	y.	0.0316
g.	0.0660					z.	0.0430

•The correlation should be maximum when the key k translates the ciphertext into English; in fact $k=3$ yields the right message: **HELLO WORLD**

2. Cryptographic Systems

Definition

A cryptographic system (or *cryptosystem*) has five components:

1. A plaintext space, M .
2. A ciphertext message space, C .
3. A key space, K .
4. A family of enciphering transformations, $E_k: M \rightarrow C$, where $k \in K$.
5. A family of deciphering transformations, $D_k: C \rightarrow M$, where $k \in K$.

For a given key k , D_k is the inverse of E_k : $\forall m \in M \bullet D_k(E_k(m)) = m$.

Properties

-Secrecy: It should be computationally infeasible for a cryptanalyst to systematically determine the deciphering transformation D_k from intercepted ciphertext c , even if the corresponding plaintext m is known.

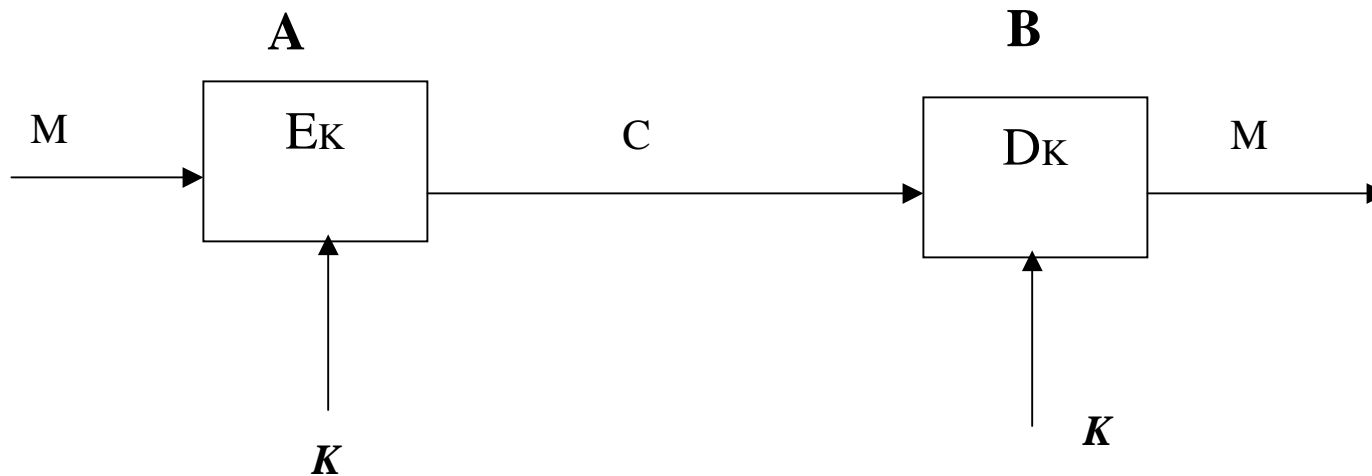
-Authenticity: It should be computationally infeasible for a cryptanalyst to systematically determine the enciphering transformation E_k given c , even if the corresponding plaintext m is known.

Available Cryptosystems

-There are two kinds of cryptosystems: *symmetric* cryptosystems and *asymmetric* cryptosystems.

Symmetric cryptosystems:

- The same key is used for enciphering and deciphering.



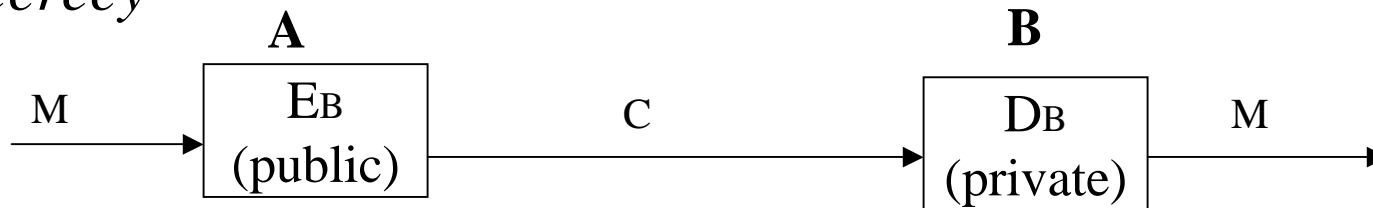
- Secrecy and authenticity can be achieved in such case only by protecting both Ek and Dk.

Asymmetric (Public) Cryptosystems

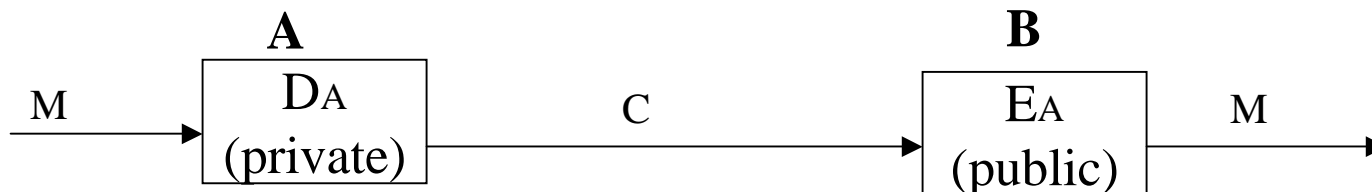
-Concept introduced in 1976 by Diffie and Hellman:

- The enciphering and deciphering keys differ in such a way that at least one key is computationally infeasible to determine from the other.
- Thus one of the transformations E_k or D_k can be revealed without endangering the other.
- Each user has both a public key and a private key, and two users can communicate knowing only each other's public keys.
- Protecting the separate transformation- D_k for secrecy, E_k for authenticity, provides secrecy and authenticity.

Secrecy



Authenticity



Secrecy and Authenticity

-Only few available public-key systems can be used for both authenticity and secrecy; an example of such cryptosystem is RSA.

- To use a public-key system for both secrecy and authenticity, the ciphertext space C must be equivalent to the plaintext space M so that any pair of transformations E and D can operate on both plaintext and ciphertext.

Secrecy	Authenticity	Both
$E: M \rightarrow C$	$D: M \rightarrow C$	$E: M \rightarrow M$
$D: C \rightarrow M$	$E: C \rightarrow M$	$D: M \rightarrow M$
$D(E(M)) = M$	$E(D(M)) = M$	$D(E(M)) = M$
		$E(D(M)) = M$

Digital Signatures

-Property private to a user or a process that is used for signing messages.

-Let B be the recipient of a message M signed by A; then A's signature must satisfy the following requirements:

- 1. B must be able to validate A's signature on M*
- 2. It must be impossible for anyone, including B, to forge A's signature*
- 3. In case A should disavow signing a message M, it must be possible for a judge or third party to resolve a possible dispute.*

-A digital signature establishes sender authenticity. Public-key authentication systems provide a simple scheme for implementing digital signatures:

- 1. A signs M by computing $C=DA(M)$.*
- 2. B validates A's signature by checking that $EA(C)$ restores M.*
- 3. A judge resolves a possible dispute by checking whether $EA(C)$ restores M in the same way as B.*

Public-key Certificates

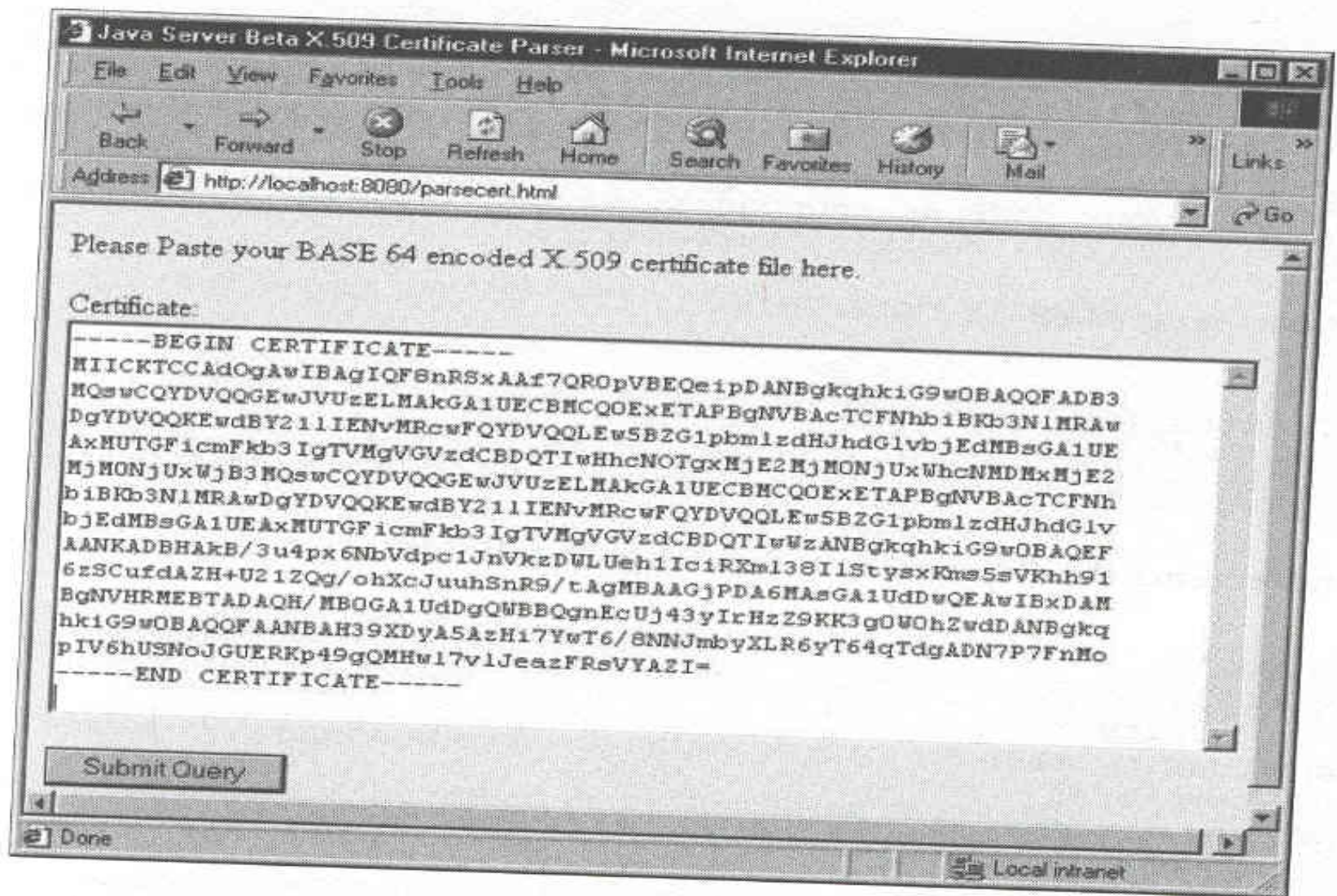
- A *certificate* is a document containing the public key of a third party and signed by a *Certification Authority (CA)*.
- Certification Authority (CA)*: A trusted third party that ensures that the holder of a private key is truly who he claims to be.

- Each certification authority issues a *certification service practices statement* that describes the process that they follow to decide if a key is genuinely the property of a third party.
- Users base themselves on this document to decide whether or not to trust the holder of the certificate in their transactions.

- The public key of a certification authority is available as a certificate issued by itself or by another CA, and called *root certificate*.

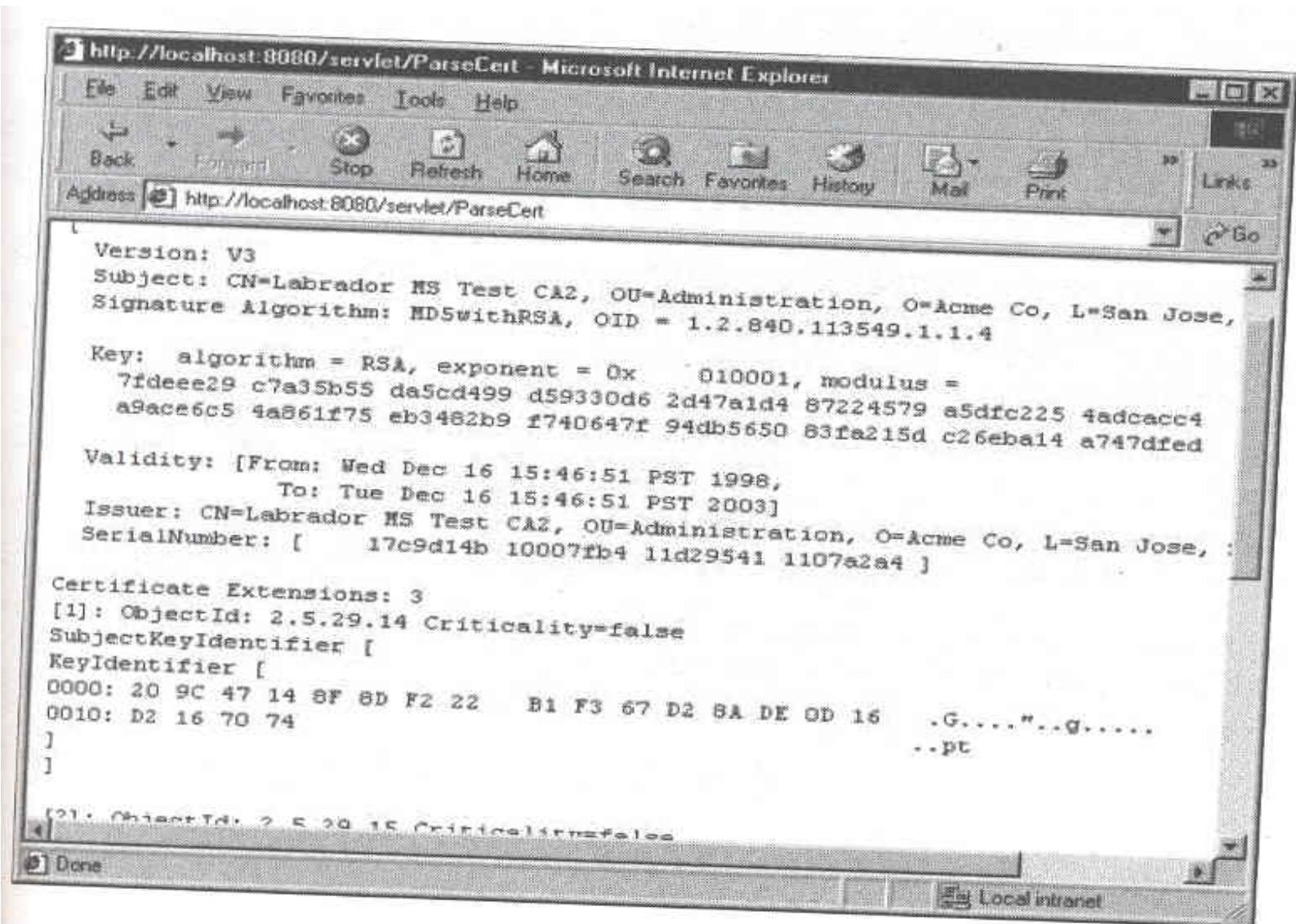
Example of Certificate

-Encrypted version



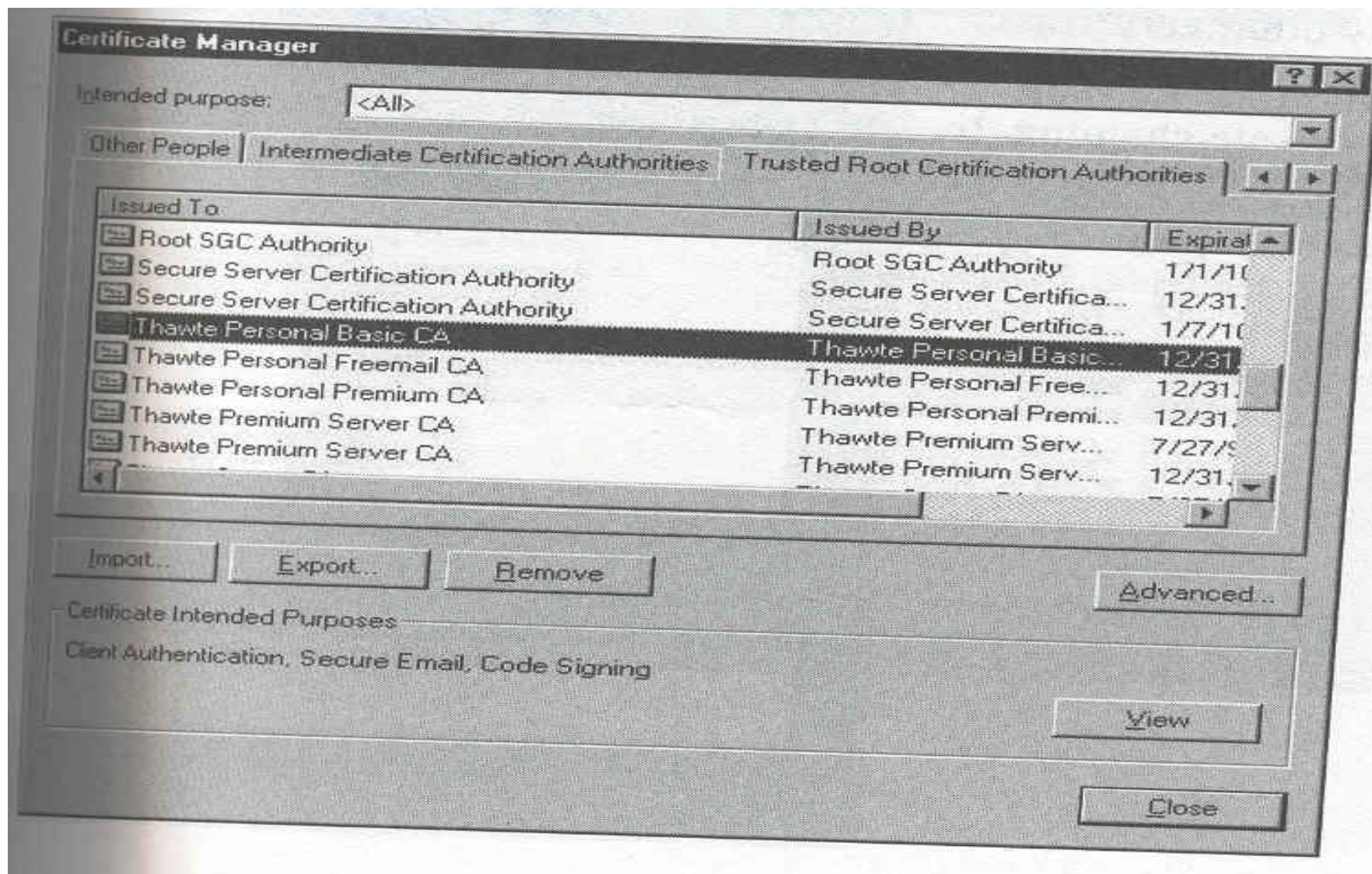
pastes the certifi

- Plain version

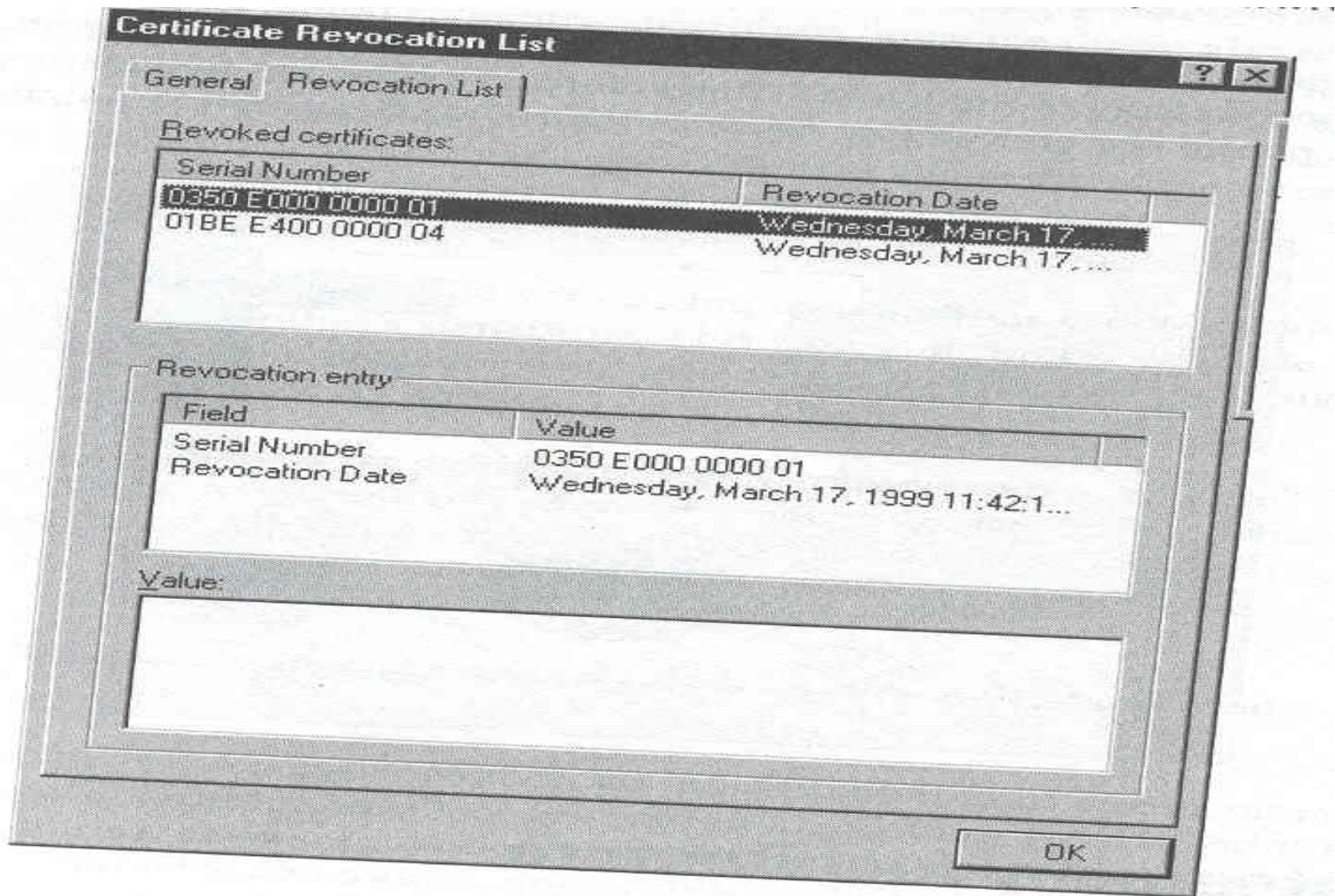


Certificate Management Tools

- Certificate Manager



-Certificate Revocation List (CRL)



3. Security of a Cryptosystem

Entropy and Uncertainty

Definition: The *Entropy* of a message M , denoted $H(M)$, is the amount of information in the message. It corresponds to the minimum number of bits needed to encode all possible meanings of the message, assuming all messages are equally likely.

Let X_1, \dots, X_n be n possible messages occurring with probabilities $p(X_1), \dots, p(X_n)$, where $\sum_{\{1 \leq i \leq n\}} p(X_i) = 1$; the entropy of a given message X is defined by the weighted average:

$$H(X) = - \sum_{\{1 \leq i \leq n\}} p(X_i) \log_2 p(X_i)$$

Examples:

What is the entropy of the following fields of a database:

-“days of the week”

-“gender”

Definition: the *uncertainty* of a message corresponds to the number of plaintext bits that must be recovered when the message is scrambled in ciphertext in order to learn the plaintext. The uncertainty of a message is measured by its entropy.

Equivocation:

-The uncertainty of a message can be reduced given additional information.

Given a message Y in the set $\{Y_1, \dots, Y_n\}$, where $(\sum_{1 \leq i \leq n} p(Y_i) = 1)$, let $p_Y(X)$ be the conditional probability of message X given message Y (e.g. $p(X/Y)$), and let $p(X, Y)$ be the joint probability of message X and message Y , thus:

$$p(X, Y) = p_Y(X)p(Y)$$

Definition: the *equivocation* is the conditional entropy of X given Y :

$$\begin{aligned} H_Y(X) &= - \sum_{\{X, Y\}} p(X, Y) \log_2 p_Y(X) \\ &= - \sum_{\{Y\}} p(Y) \sum_{\{X\}} p_Y(X) \log_2 (p_Y(X)) \end{aligned}$$

Rate of a Language

-According to Shannon the entropy depends on the length of the text.

-The rate of a given language, for a message M of length N is measured by:

$$r = H(M)/N$$

- Corresponds to the average number of bits of information per character.
- *Example*: English language: r varies between 1.0bits/letter and 1.5 bits/letter for large values of N .

Definition: the *absolute rate* of a language is the maximum number of bits that can be coded in each character, assuming each character sequence is equally likely (e.g. maximum entropy of individual characters).

-For a given language containing L characters, the absolute rate is: $R = \log_2 L$

- For English language $R=4.7$ bits/letter; so the actual rate is less than the absolute one; this is because English like many other languages is redundant.

Redundancy: the redundancy of a language can be measured by taking the difference between the absolute rate and the rate. $D = R - r$

Perfect Secrecy

- A cryptanalyst may be able to break a cryptosystem by determining the key K or the plaintext P , or probabilistic information about K or P .

Definition: A cryptosystem achieves **perfect secrecy**, when the ciphertext yields no possible information about the plaintext (except possibly its length).

According to Shannon, perfect secrecy is possible only if the number of possible keys is at least as large as the number of possible messages.

- The secrecy of a cipher is measured in terms of the key equivocation $H_c(K)$ of a key K for a given ciphertext C ; that is the amount of uncertainty in K given C :

$$H_c(K) = -\sum_{\{C\}} p(C) \sum_{\{K\}} p_c(K) \log_2 (p_c(K))$$

Definition: the entropy of a cryptosystem is obtained using the size of the keyspace K : **$H(K) = \log_2 K$**

(correspond to the key entropy or number of bits in the key assuming that all keys are equally likely)

-The unicity distance corresponds to the amount of text necessary to break a cipher;

Definition: The **unicity distance** corresponds to the amount of ciphertext needed to uniquely determine a key. That is measured by the smallest N such that $H_c(K)$ is close to 0.

$$N = H(K)/D$$

(D is the redundancy).

•*Example:* calculate the unicity distance for the DES cryptosystem, which is a 64-bit system.

$$N_{\text{DES}} = H(K)/D = 64/3.2 = 20 \text{ characters}$$

Algorithm Complexity

- The strength of a cryptosystem can be measured by analyzing the computational complexity of cryptographic techniques and algorithms
- Complexity correspond to the computational power needed to execute the algorithm.
- Complexity is measured by two variables: T (*for time complexity*) and S (*for space complexity, or memory requirements*).
- Both T and S are commonly expressed as function of n the size of the input, using the “*big O*” notation, which measures the order of magnitude of the computational complexity.
- O corresponds to the term of the complexity function, which grows the fastest as n gets larger.

Example: if the time complexity is $4n^3 + 8n + 5$, then the computational complexity is $O(n^3)$.

Categories:

- $O(n^m)$ with m a constant: *polynomial* algorithms
 - m=0 -> constant; m=1: linear; m=2 -> quadratic; m=3 -> cubic etc.
 - $O(t^{f(n)})$ where t is a constant greater than 1, and f(n) is a polynomial: *exponential*
- Consider, for instance, a machine capable of performing one instructions/microseconds → 10^6 instructions per second:

Class	Complexity	Number of operations For n=10 ⁶	Real Time
Constant	O(1)	1	1μsec
Linear	O(n)	10 ⁶	1 sec
Quadratic	O(n ²)	10 ¹²	10 days
Cubic	O(n ³)	10 ¹⁸	27, 393 years
Exponential	O(2 ⁿ)	10 ³⁰¹⁰³⁰	10 ³⁰¹⁰¹⁶ days

Brute Force Attack:

-Many ciphers can be solved by exhaustively searching the entire key space, trying each possible key.

- If $n=2^{H(K)}$ is the size of the key space, then the running time of this strategy is: $T= O(2^{H(K)})$.
- Thus, the time is linear in the number of keys but exponential in the key length.
- So doubling, for instance, the length of the keys used in DES from 56 to 112 bits can have a dramatic impact on the difficulty of breaking the cipher.

4. Transposition Ciphers

-Transposition ciphers rearrange characters according to some scheme.

- Many transposition ciphers permute the characters of the plaintext with a fixed period d .

-A transposition cipher can be defined by providing an integer d , and a permutation $f: \mathbb{Z}_d \rightarrow \mathbb{Z}_d$ (where \mathbb{Z}_d is the set of integers 1 through d)

- The key: $K = (d, f)$

- A plaintext message M is enciphered as follows:

$$M = m_1 \dots m_d m_{d+1} \dots m_{2d} \dots$$



$$E_k(M) = m_{f(1)} \dots m_{f(d)} m_{d+f(1)} \dots m_{d+f(d)} \dots$$

Example: encryption

suppose $d=4$, and

f gives the permutation:

i:	1	2	3	4
f(i):	2	4	1	3

M= RENAISSANCE

Compute $E_k(M)$?

Example: Cryptanalysis

-A transposition may be subject to successful cryptanalysis because the relative frequencies of the letters in the ciphertext can closely match expected frequencies for plaintext.

- *Assuming all keys are equally likely, what is the the entropy of the key of a transposition cipher with a period d ?*
- *Determine the expected number N of characters required to break the cipher, for a period $d=27$?*

5. Substitution Ciphers

-There are several kinds of substitution ciphers: simple, homophonic, polyalphabetic, and polygram substitution ciphers.

Simple Substitution Ciphers

-Simple one-to-one mapping is used to encipher an entire message.

-Let: A and C be n -character alphabets

$f: A \rightarrow C$ a one-to-one mapping

$$A = \{a_0, \dots, a_{n-1}\}$$

$$C = \{f(a_0), \dots, f(a_{n-1})\}$$

A plaintext M is enciphered as follows:

$$M = m_1 \dots m_p$$



$$E_k(M) = f(m_1) \dots f(m_p)$$

Example: ciphers based on shifted alphabet

-f is defined by $f(a) = (a+k) \bmod n$ where n is the size of the alphabet and a denotes both a letter and its position in A .

- A is given as follows:

0-A	7-H	13-N	20-U
1-B	8-I	14-O	21-V
2-C	9-J	15-P	22-W
3-D	10-K	16-Q	23-X
4-E	11-L	17-R	24-Y
5-F	12-M	18-S	25-Z
6-G		19-T	

Compute $E_k(M)$ for $M=RENAISSANCE$, and $k=3$

Example: cryptanalysis

Assuming that all keys are equally likely, how many characters are needed to break the above cipher?

Homophonic Substitution Ciphers

-Use a one-to-many mapping; each plaintext character can be mapped into a ciphertext element picked at random from a set of characters.

-Let: A, C : n -character alphabet

$f: A \rightarrow \wp(C)$, a one-to-many mapping

A plaintext message M is enciphered as follows:

$M = m_1 m_2 \dots$



$E_k(M) = c_1 c_2 \dots$, where $c_i \in f(m_i)$

Example: Suppose that the English letters are enciphered as integers between 0 and 99.

Consider the following assignment of integers to letters:

<u>Letter</u>	<u>Homophones</u>
A	17 19 34 56 60 67 83
I	08 22 53 65 88 90
L	03 44 76
N	02 09 15 27 32 40 59
O	01 11 23 28 42 54 70 80
P	33 91
T	05 10 20 29 45 58 64 78 99

Compute $E_k(M)$ for $M=PLAIN PILOT$

Polyalphabetic Substitution Ciphers

- Use multiple mappings from plaintext to ciphertext characters; the mappings are usually one-to-one as in simple substitution.
- Most polyalphabetic substitution ciphers are periodic substitution ciphers based on a period d .
- Let: C_1, \dots, C_d , cipher alphabets
 - $f_i: A \rightarrow C_i$, mapping from plaintext alphabet A to the i th cipher alphabet C_i ($1 \leq i \leq d$)

A plaintext message M is enciphered as follows:

$$M = m_1 \dots m_d m_{d+1} \dots m_{2d} \dots$$



$$E_k(M) = f_1(m_1) \dots f_d(m_d) f_1(m_{d+1}) \dots f_d(m_{2d}) \dots$$

Example: Vigenere cipher

The key K is specified by a sequence of letters $K=k_1\dots k_d$, where k_i ($i=1,\dots,d$) gives the amount of shift in the i th alphabet:

$$f_i(a) = (a+k_i) \bmod n.$$

Compute $E_k(M)$, for $M=RENAISSANCE$, and $K=BAND$

Example: cryptanalysis

How many characters are required to break the Vigenere cipher, assuming a period d .

Example: one-time pad

-A cipher in which the key is a random sequence of characters and is not repeated; the key is only used once.

- Let $M=m_1m_2\dots$ a plaintext bit stream and $K=k_1k_2\dots$ a key bit stream, $E_k(M) = c_1c_2\dots$, where $c_i=(m_i\oplus k_i) \bmod 2$, $i=1,2,\dots$
- Because $k_i\oplus k_i=0$, deciphering is performed by: $c_i\oplus k_i=m_i\oplus k_i\oplus k_i=m_i$

Compute $E_k(M)$ for $M=A$, and $K=D$ ($A=11000$; $D=10010$)

Polygram Substitution Ciphers

- The most general forms of substitution ciphers, permitting arbitrary substitutions for groups of characters.
- Enciphering larger blocks of letters makes cryptanalysis harder by destroying the significance of single-letter frequencies.

Polygram Substitution Ciphers

- The most general forms of substitution ciphers, permitting arbitrary substitutions for groups of characters.
- Enciphering larger blocks of letters makes cryptanalysis harder by destroying the significance of single-letter frequencies.

Example: Playfair cipher

-Digram substitution cipher that uses a 5×5 matrix (J is not used) to generate the key; a pair of plaintext letters m_1m_2 is enciphered as follows:

1. If m_1 and m_2 are in the same row, then c_1 and c_2 are the two characters to the right of m_1 and m_2 , respectively, where the first column is considered to be to the right of the last column.
2. If m_1 and m_2 are in the same column, then c_1 and c_2 are the two characters below m_1 and m_2 , respectively, where the first row is considered to be below the last row.
3. If m_1 and m_2 are in different rows and columns, then c_1 and c_2 are the other two corners of the rectangle having m_1 and m_2 as corners, where c_1 is in m_1 's row and c_2 is in m_2 's row.
4. If $m_1=m_2$, a null letter (e.g., X) is inserted into the plaintext between m_1 and m_2 to eliminate the double.
5. If the plaintext has an odd number of characters, a null letter is appended to the end of the plaintext.

Compute $E_k(M)$, for $M=RENAISSANCE$ with K :

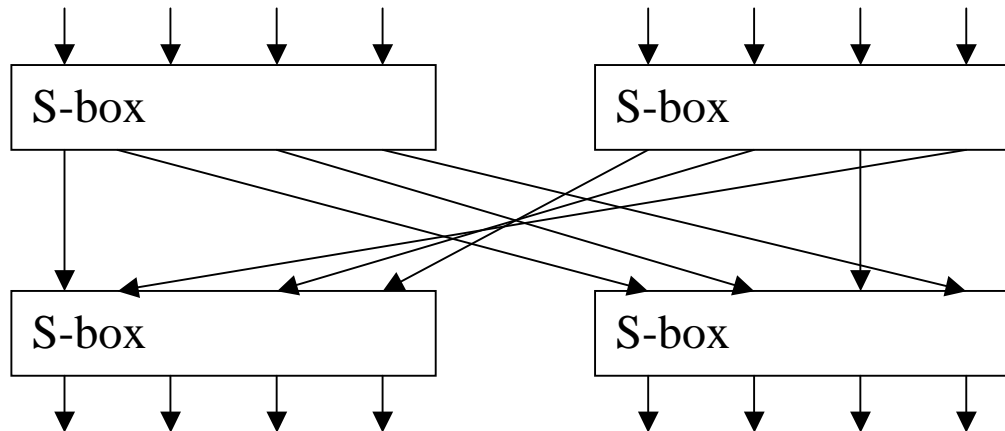
```
H A R P S  
I C O D B  
E F G K L  
M N Q T U  
V W X Y Z
```

6. Product Ciphers

Substitution-Permutation Ciphers

Algorithm design

- Shannon proposed to design strong ciphers by mixing different kinds of transformations. This can be achieved by alternating *substitutions* and *transpositions*.
- The earliest block ciphers were based on this principle and so were called *SP-networks*.



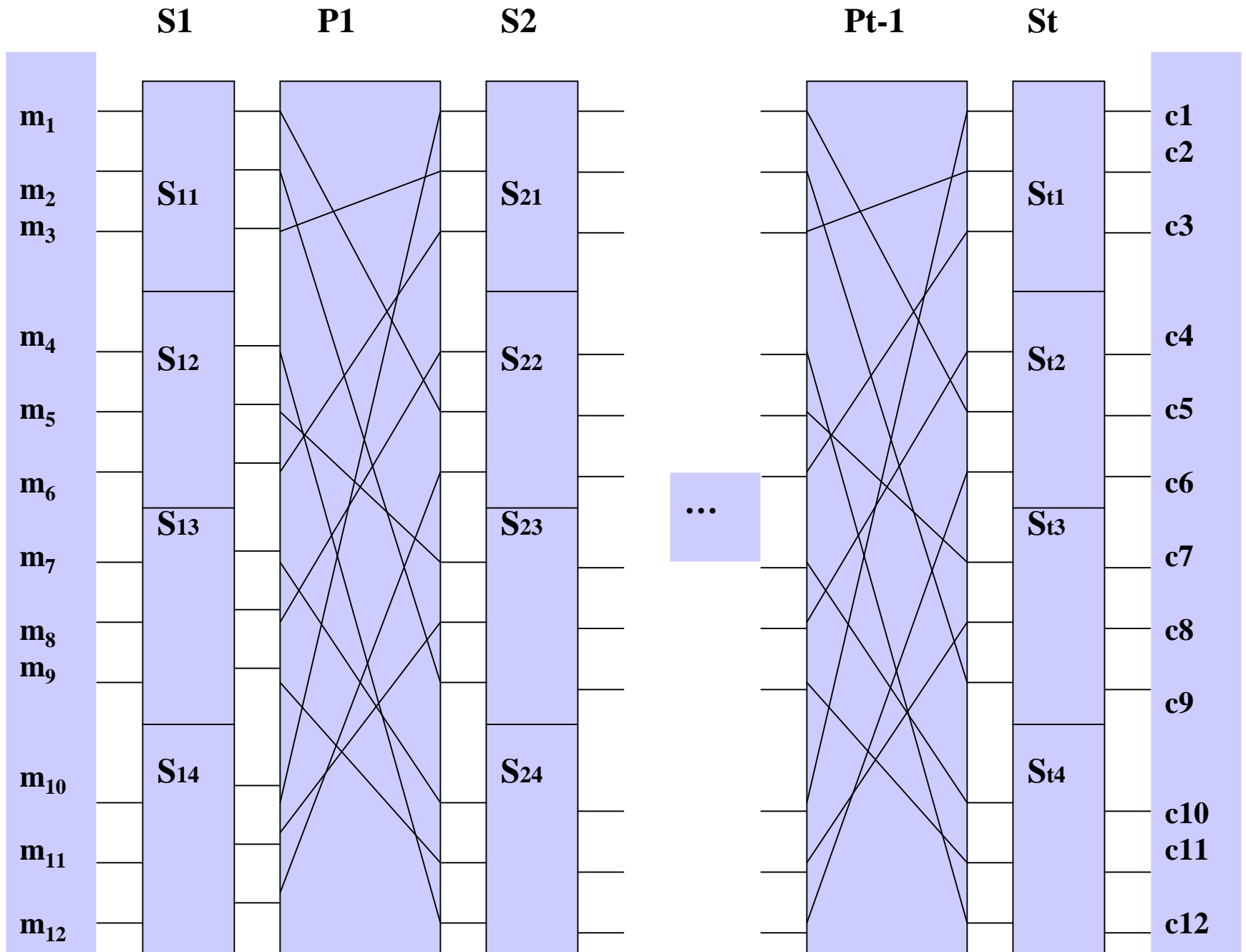
-Three things need to be done to make the algorithm design secure:

1. The cipher needs to be “wide” enough.
2. The cipher needs to have enough rounds.
3. The S-boxes need to be suitably chosen.

Example: LUCIFER cipher

$$C = Ek(M) = S_t \circ P_{t-1} \circ \dots \circ S_2 \circ P_1 \circ S_1(M)$$

- Each S_i is a function of the key K , and is broken into 4 smaller substitutions S_{i1}, \dots, S_{i4} , operating on a 3-bit sub-block to reduce the complexity of the circuits



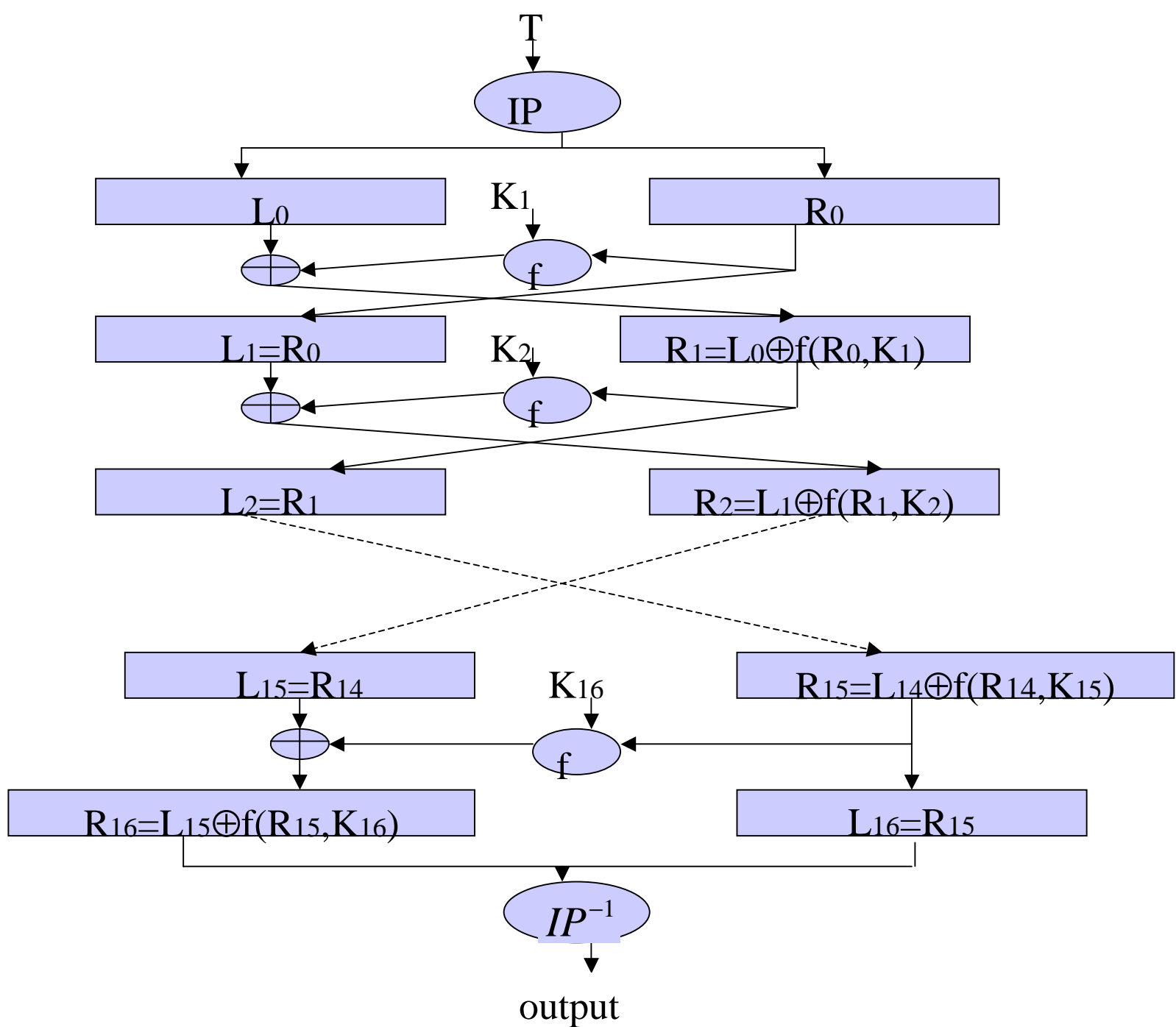
Digital Encryption Standard (DES)

- DES has been created in 1977 at IBM, as an outgrowth of LUCIFER.
- DES enciphers 64-bit blocks of data with a 56-bit key, and has been implemented in both hardware and software.
- The same algorithm is used for encryption and decryption.
 - An input block T is first passed through a permutation IP .
 - Then the output of the initial permutation is submitted to 16 iterations of a function f (substitution + transposition) .
 - Finally the inverse permutation IP^{-1} is applied, which gives the final result.

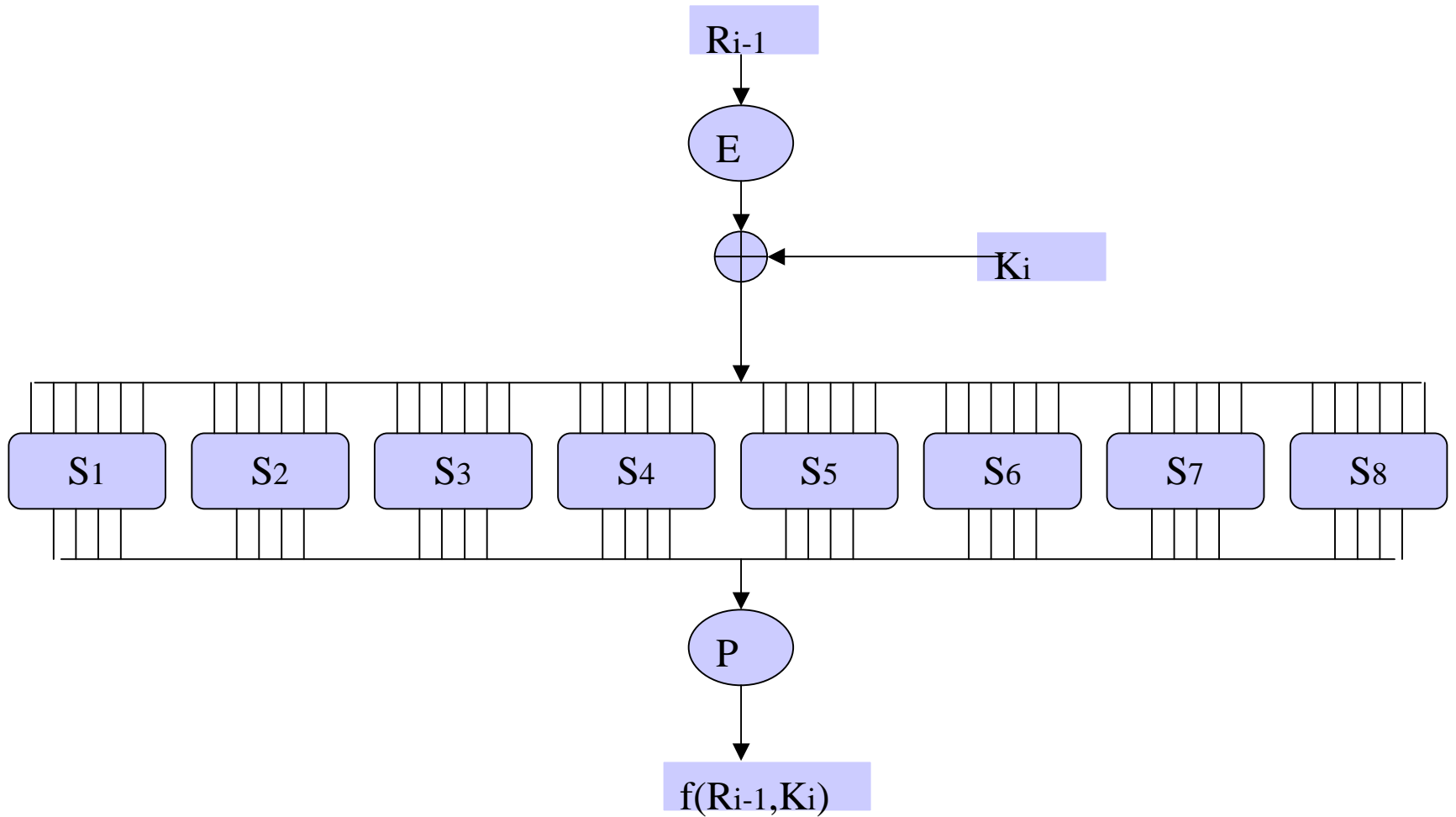
Let $T_i = L_i R_i$ denotes the result of the i^{th} iteration, with L_i and R_i the left and right halves of T_i :

$$L_i = t_1 \dots t_{32}, R_i = t_{33} \dots t_{64} \Rightarrow L_i = R_{i-1}, R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

where K_i is a 48-bit key.



Calculation of $f(R_{i-1}, K_i)$



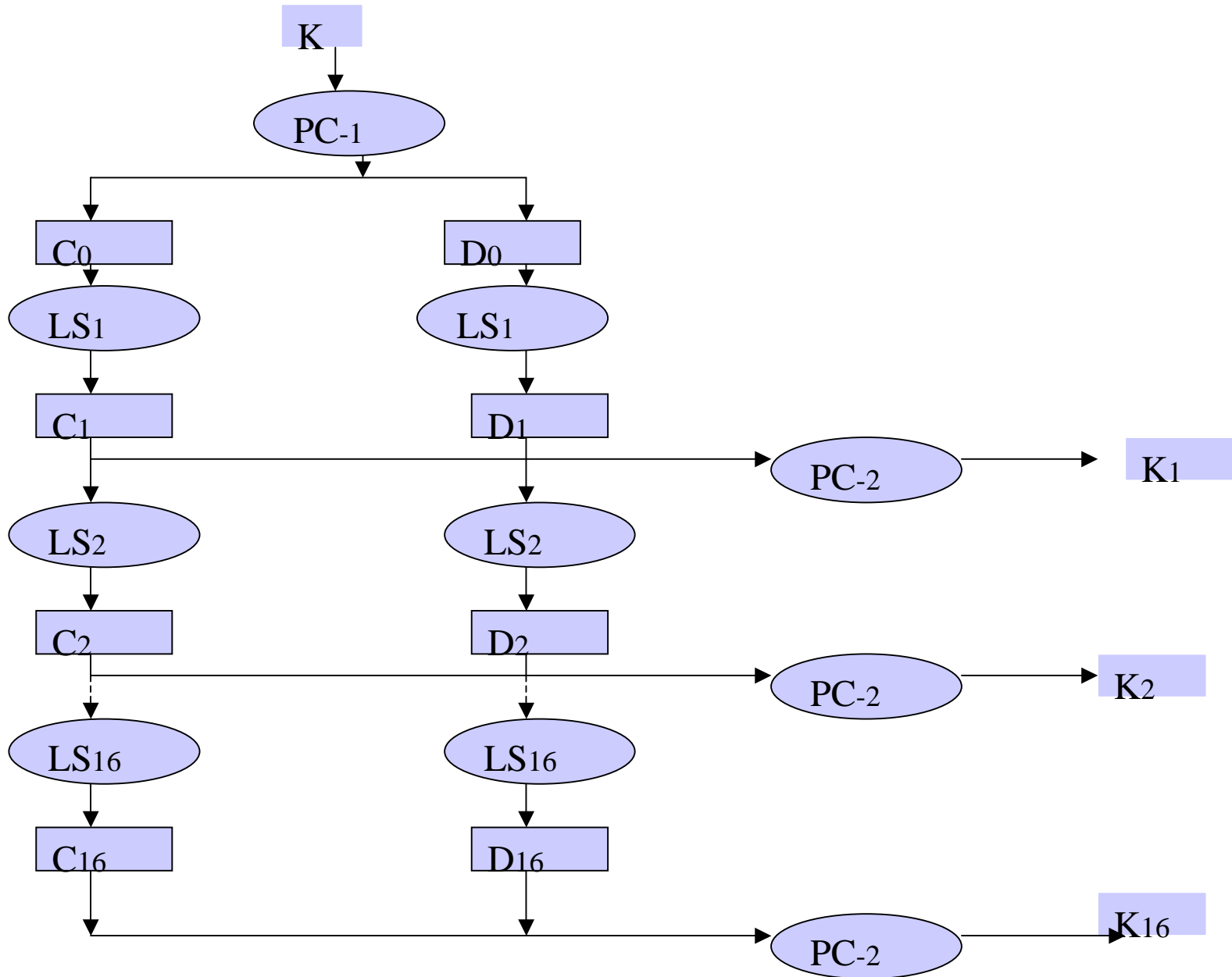
1. R_{i-1} is first expanded in 48 bits using a permutation E ,
2. $E(R_{i-1}) \oplus K_i$ is taken and divided into 8 blocks of 6 bits:
$$E(R_{i-1}) \oplus K_i = B_1 \dots B_8, \text{ where } B_i = b_1 \dots b_6$$
3. Each block B_i is passed through a *S-box*, returning a 4-bit block $S_i(B_i)$
4. The $S_i(B_i)$ are then concatenated and transposed using a permutation P : $f(R_{i-1}, K_i) = P(S_1(B_1) \dots S_8(B_8))$

S-box:

An input block $B_i = b_1 \dots b_6$ is transformed using a substitution table:

- The integer corresponding to $b_1 b_6$ specifies a row number,
- The integer corresponding to $b_2 b_3 b_4 b_5$ specifies a column.
- $S_i(B_i)$ is the 4-bit representation of the integer in that row and column.

Key Calculation



- A different key K_i derived from K is used for each iteration.
- K is input as a 64-bit block, with 8 parity bits in positions 8, 16,...,64
- Then a permutation PC_{-1} is applied discarding the parity bits and transposing the remaining 56 bits.
- The results $PC_{-1}(K)$ is then split into two halves C and D of 28 bits each.
- A key K_i is derived by successively shifting left C and D for each iteration:

$$C_i = LSi(C_{i-1}), D_i = LSi(D_{i-1})$$

$$K_i = PC_{-2}(C_i D_i),$$

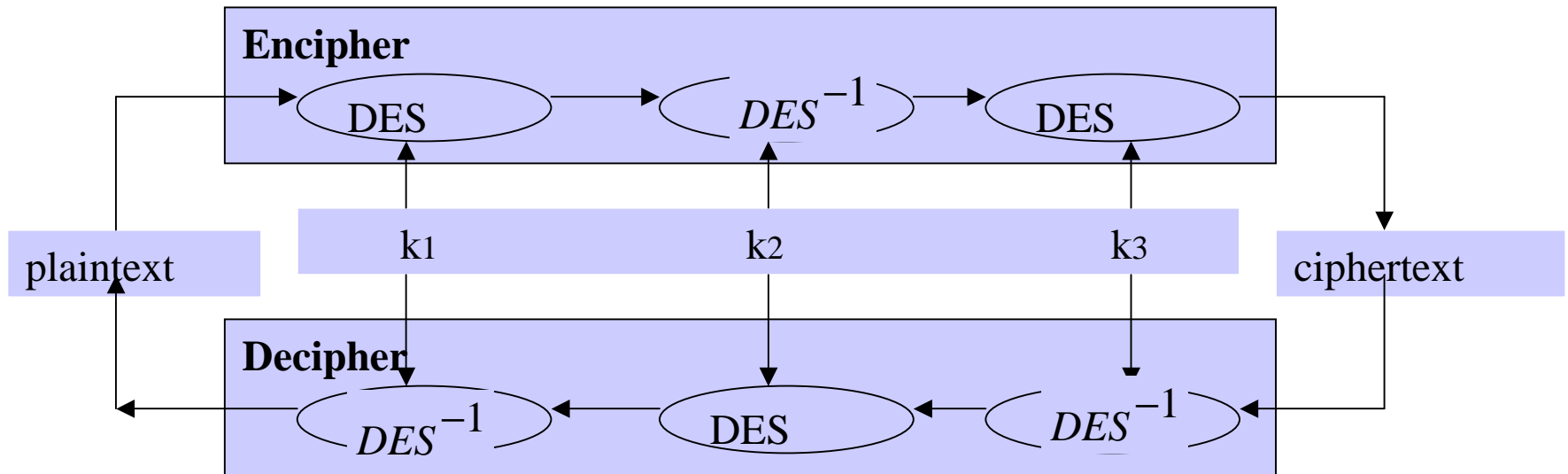
where LS_i is a left circular shift by specified number of positions, and PC_{-2} is a permutation.

Triple DES

-Since its invention, several weaknesses have been identified in DES.

- Key size is one of them: 56-bits keys are vulnerable to known plaintext attack by exhaustive search.

-Solutions: either increase the size of the key (->112 bits) or use a multiple encryption scheme → *Triple DES*.



- A plaintext message M is encrypted, decrypted, and then encrypted using different keys: $C = DES_{k3}(DES_{k2}^{-1}(DES_{k1}(M)))$

- The message is restored using the reverse operation: $C = DES_{k1}^{-1}(DES_{k2}(DES_{k3}^{-1}(C)))$

7. Exponentiation Ciphers

Mathematical Background

Fermat's Little Theorem:

if m is a prime, and a is not a multiple of m then: $a^{m-1} \bmod m = 1$

Euler Totient Function:

-The reduced set of residues mod n is the subset of the complete set of residues which are relatively prime to n .

Example:

for $n=12 \Rightarrow \{1,5,7,11\}$;

if n prime $\Rightarrow \{1, \dots, n-1\}$

-The Euler Totient function denoted $\Phi(n)$ is the cardinality of the reduced set of residues mod n .

•If n is prime then $\Phi(n) = n-1$;

•if $n = pq$, where p and q are primes then $\Phi(n) = (p-1)(q-1)$

Euler's generalization of Fermat's little theorem:

If $\gcd(a,n) = 1$ then $a^{\Phi(n)} \bmod n = 1$

Exponentiation Ciphers

-Enciphering and deciphering are based on Euler's generalization of Fermat's Theorem, which states that for every M relatively prime to n :

$$M^{\phi(n)} \bmod n = 1 \quad (1)$$

Theorem 1: Given e and d satisfying $ed \bmod \Phi(n) = 1$ and a message $M \in [0, n-1]$ such that $\gcd(M, n) = 1$,

$$(M^e \bmod n)^d \bmod n = M$$

-Cipher:

- Encipher a message block $M \in [0, n-1]$ by computing the exponential

$$C = M^e \bmod n \quad (2)$$

where $K = (e, n)$ is the encryption key.

- M is restored by the same operation, but using a different exponent d .

$$M = C^d \bmod n \quad (3)$$

-By symmetry, enciphering and deciphering are commutative and mutual inverses:

$$(M^d \bmod n)^e \bmod n = M^{de} \bmod n = M \quad (4)$$

Rivest, Shamir, and Adelman (RSA) Algorithm

-Exponentiation cipher based on the use of the product of two very large prime numbers (greater than 10^{100}), and the fact that the computation of large prime factors is difficult.

-To find a key pair e, d :

1. Choose two large prime numbers, P and Q (each greater than 10^{100}), and form $N = P \times Q$, $Z = (P-1) \times (Q-1)$
2. For d choose any number that is relatively prime with Z
3. To find e solve the equation: $e \times d \bmod Z = 1$
4. Divide the plaintext into equal blocks of length k bits where $2^k < N$
(in practice k is in the range 512-1024)
5. A single block of plaintext is *encrypted* using $E(e, N, M) = M^e \bmod N$
6. A block of encrypted text C is *decrypted* using $D(d, N, M) = C^d \bmod N$

- RSA can be used both for secrecy and authenticity in a public-key system due to the symmetry (Eq. (4)) inherent to exponential ciphers.

8. Cryptography based on Discrete Logarithms

-There are 2 flavours of algorithms based on discrete logarithm: arithmetic and elliptic curves.

-The arithmetic approach is based on the difficulty of finding, given a large prime number p , the discrete logarithm of a number y :

$$y = g^x \bmod p$$

-The mapping $f: x \rightarrow y = g^x \bmod p$ is a one-way function, with the additional properties that:

$$f(x+y) = f(x)f(y) \text{ and } f(nx) = (f(x))^n$$

The Diffie-Hellman Protocol

-Public key encryption scheme based on a commutative encryption function.

1. *Alice encrypts message M with her key: $ka \rightarrow \{M\}_{ka}$.*
2. *Alice sends $\{M\}_{ka}$ to Bob.*
3. *Bob in his turn encrypts the received message: $\rightarrow \{\{M\}_{ka}\}_{kb}$*
4. *Bob sends $\{\{M\}_{ka}\}_{kb}$ back to Alice.*
5. *Alice is able to decrypt the received message due to **commutativity** $\{\{M\}_{ka}\}_{kb} = \{\{M\}_{kb}\}_{ka}$: $\rightarrow \{M\}_{kb}$*
6. *Alice sends $\{M\}_{kb}$ to Bob, who can decrypt it using his key $kb \rightarrow M$.*

-Diffie and Hellman use a commutative encryption function based on discrete logarithm:

Appropriate prime p and generator g are chosen, and common for all users.

1. Alice chooses a secret random number x_a (\rightarrow her private key) and publish $y_a = g^{x_a}$ (her public key).
2. Bob does the same with x_b secret and $y_b = g^{x_b}$ public.
3. Alice uses $y_b^{x_a} = g^{x_a x_b}$ to encrypt a message to Bob.
4. Bob uses $y_a^{x_b} = g^{x_a x_b}$ to decrypt the received message.

- In practice Alice and Bob uses $g^{x_a x_b}$ as shared session key for their communication, and may use any desired secret cipher.
- The basic protocol itself doesn't provide forward security; it is easily subject to middleperson attacks and so on. This can be dealt with by authenticating the participants (e.g. digital signatures).

9. Advanced Encryption Standard (AES)

Historical Background

- Second AES conference in March 1999
- Selected five candidates:
 - MARS
 - RC6
 - Rijndael
 - Serpent
 - Twofish.
- In October 2000 Rijndael is selected by NIST as proposed AES

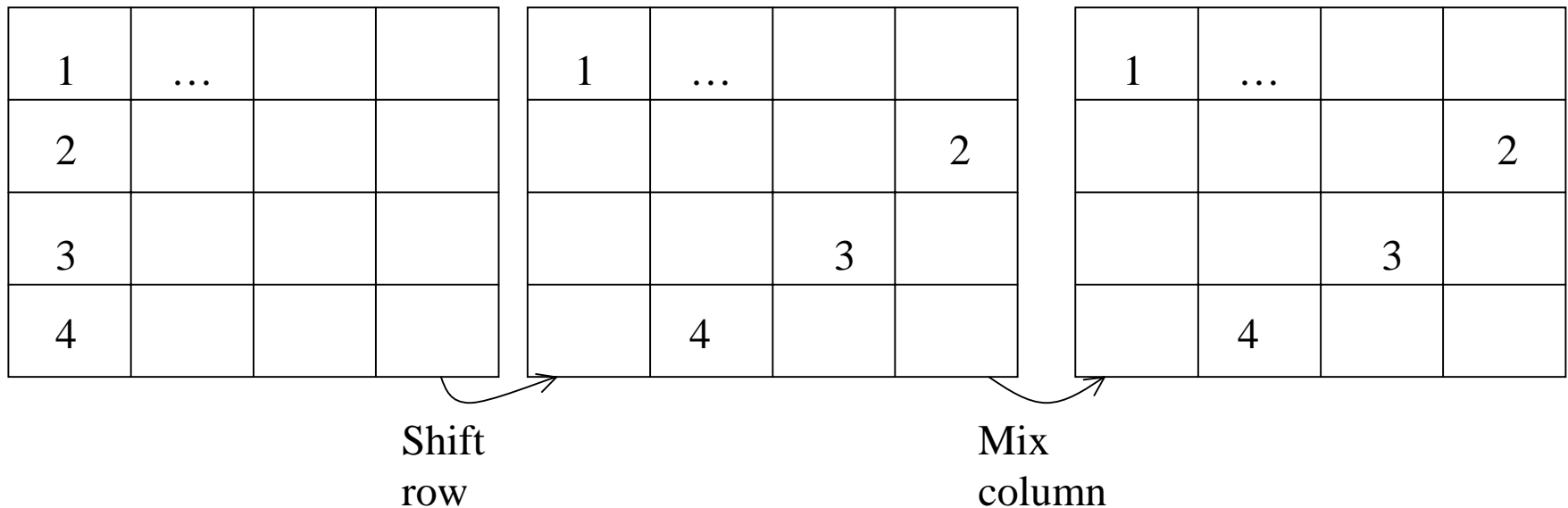
The Algorithm

- Invented by Vincent Rijmen and Joan Daemen, and adopted recently after a competition organized by NIST, as US standard.
- Acts on 128 bit blocks and can use a key of 128, 192, or 256 bits in length.
- Based on an SP-network that uses a single S-box which acts on a byte input to give a byte output.

	Plaintext block Size (bits)	Number of rounds
AES-128	128	10
AES-192	128	12
AES-256	128	14

- The S-box is defined by: $S(x) = M(1/x) + b$ over the field $GF(2^8)$ where M is a suitably chosen matrix and b is a constant.
- The linear transformation, between the rounds, is based on arranging the 16 bytes of the value enciphered in a 4×4 matrix and then doing bitwise shuffling and mixing operations:

1. *The first step is the shuffle: the top row of four bytes is left unchanged, while the second row is shifted one place to the left, the third row by two places and the fourth row by three places.*
- *The second step is the column mixing: 4 bytes in a column are mixed using matrix multiplication.*



- The key material is added byte by byte after the linear transformation (16bytes/round)

10. Cryptographic Techniques

Block and Stream Ciphers

Block cipher

-Breaks a plaintext message M into successive blocks $M1, M2, \dots$, and encrypts them using the same key: $E_k(M) = E_k(M1)E_k(M2)\dots$

$M = M1M2M3\dots$



$C = E_k(M1)E_k(M2)E_k(M3)\dots$

Stream Cipher

-Breaks the message into successive characters m_1, m_2, \dots and encrypts each character m_i using the i th element of a key stream $K = k_1 k_2 \dots$

$$Ek(M) = Ek_1(m_1)Ek_2(m_2)\dots$$

•Most stream ciphers use a simple xor for encryption/decryption:

$$E_{k_i}(m_i) = m_i \oplus k_i$$

$$D_{k_i}(c_i) = E_{k_i}(m_i) \oplus k_i = (m_i \oplus k_i) \oplus k_i = m_i$$

-A stream cipher can be *periodic* or *nonperiodic*. In a periodic stream cipher, the key stream repeats after d characters.

-There are two types of stream ciphers: *synchronous* and *self-synchronous*:

- Synchronous stream cipher*: the key stream is generated independently of the message stream.
- Self-synchronous stream cipher*: each key character is derived from a fixed number n of preceding ciphertext character.

Modes of Operation

-The mode of operation specifies how a block cipher with a fixed block size (e.g. 8 bytes for DES, 16 for AES) can be extended to process messages of arbitrary length.

Electronic Code Book (ECB)

-Each succeeding block of plaintext is encrypted with the block cipher to get ciphertext.

- ECB is adequate for simple operations such as challenge-response and some key management tasks (e.g. encrypt PIN in cash machines) etc.
- It is inadequate for redundant data encryption, or messages of more than one block in which there are some authenticity requirement (e.g. the patterns will appear quickly).

Cipher Block Chaining (CBC)

-CBC is adequate for encrypting more than one block.

-It is intended to prevent identical portions of plaintext encrypting to identical pieces of cipher text. It is effective at disguising any patterns in the plaintext.

- Each block is XORed with the preceding ciphertext, and then encrypted; at the start of each sequence of blocks, a different piece of plaintext called *initialization vector (IV)* is inserted.

- The decryption of a block is done by decrypting it, and then XORing the result with the preceding encrypted block.

-CBC is appropriate only on reliable connection; decryption will fail if any blocks of cyphertext are lost.