Week 3: Modern Symmetric Ciphers

PhD. Ngoc-Tu Nguyen

tunn@uit.edu.vn

What is cryptograph?

- Cryptology= Cryptography + Cryptanalysis

What?

Goals

Cipher systems

- Confidentiality
- Privacy

- Sysmmetric (AES)
- Asymmetric (RSA, ECC, CRYSTALS-KYBER)

- Integrity
- Authentication
- Non-repudiation (Accountability)

Hash functions
Message authentication code (MAC)
Digital signature (digital certificate)

- Availability

Cryptanalysis on monoalphabetic cipher?

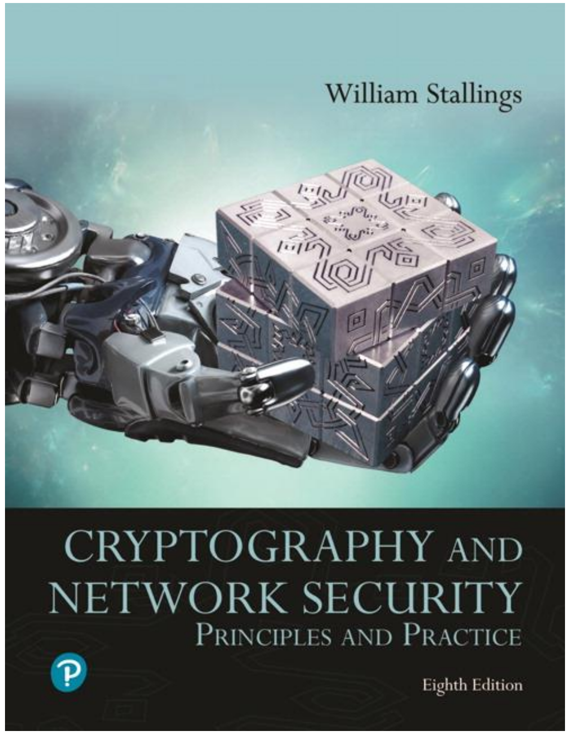
hzsrnqc klyy wqc flo mflwf ol zqdn nsoznj wskn lj xzsrbjnf, wzsxz gqv zqhhnf ol ozn glco zlfnco hnlhrn; nsoznj jnrqosdnc lj fnqj kjsnfbc, wzsxz sc xnjoqsfrv gljn efeceqr. zn rsdnb qrlfn sf zsc zlecn sf cqdsrrn jlw, wzsoznj flfn hnfnojqonb. q csfyrn blgncosx cekksxnb ol cnjdn zsg. zn pjnqmkqconb qfb bsfnb qo ozn xrep, qo zlejc gqozngqosxqrrv ksanb, sf ozn cqgn jllg, qo ozn cqgn oqprn, fndnj oqmsfy zsc gnqrc wsoz loznj gngpnjc, gexz rncc pjsfysfy q yenco wsoz zsg; qfb wnfo zlgn qo naqxorv gsbfsyzo, lfrv ol jnosjn qo lfxn ol pnb. zn fndnj ecnb ozn xlcv xzqgpnjc wzsxz ozn jnkljg hjldsbnc klj soc kqdlejnb gngpnjc. zn hqccnb onf zlejc leo lk ozn ownfov-klej sf cqdsrrn jlw, nsoznj sf crnnhsfy lj gqmsfy zsc olsrno.

Outline

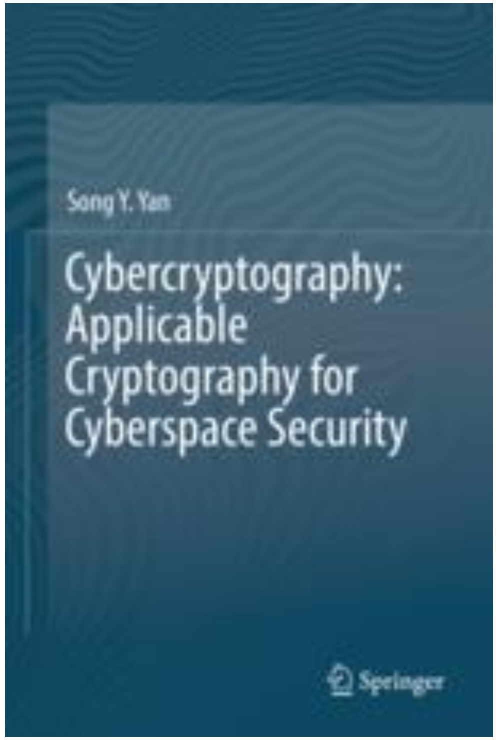
- Classical cipher algorithms (review)
- Stream Cipher
- Block cipher
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard (AES)
  - Some other ciphers
    - Searchable encryption

Textbooks and References

- Text books



[1] Chapter 4,6



[2] Chapter 5

Classical cipher algorithms

- **Substitution** Technique
  - Monoalphabetic cipher
    - Replace one character by another character
  - Polyalphabetic cipher
    - Replace some characters by other characters
      - 2 by 2:
      - 3 by 3:
- **Transposition** Technique
  - Keep the same source characters but change their positions

Monoalphabetic cipher

(1) **Caesar Cipher** (replace 1 character by 1 one character)

- Simplest and earliest known use of a substitution cipher
- Used by Julius Caesar

Key

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

plain: MEET ME AFTER THE TOGA PARTY
cipher: JBBQ JB XCQBO QEB QLDX MXOQV

Monoalphabetic cipher

(2) Generral Monoalphabetic substitution

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

EX:

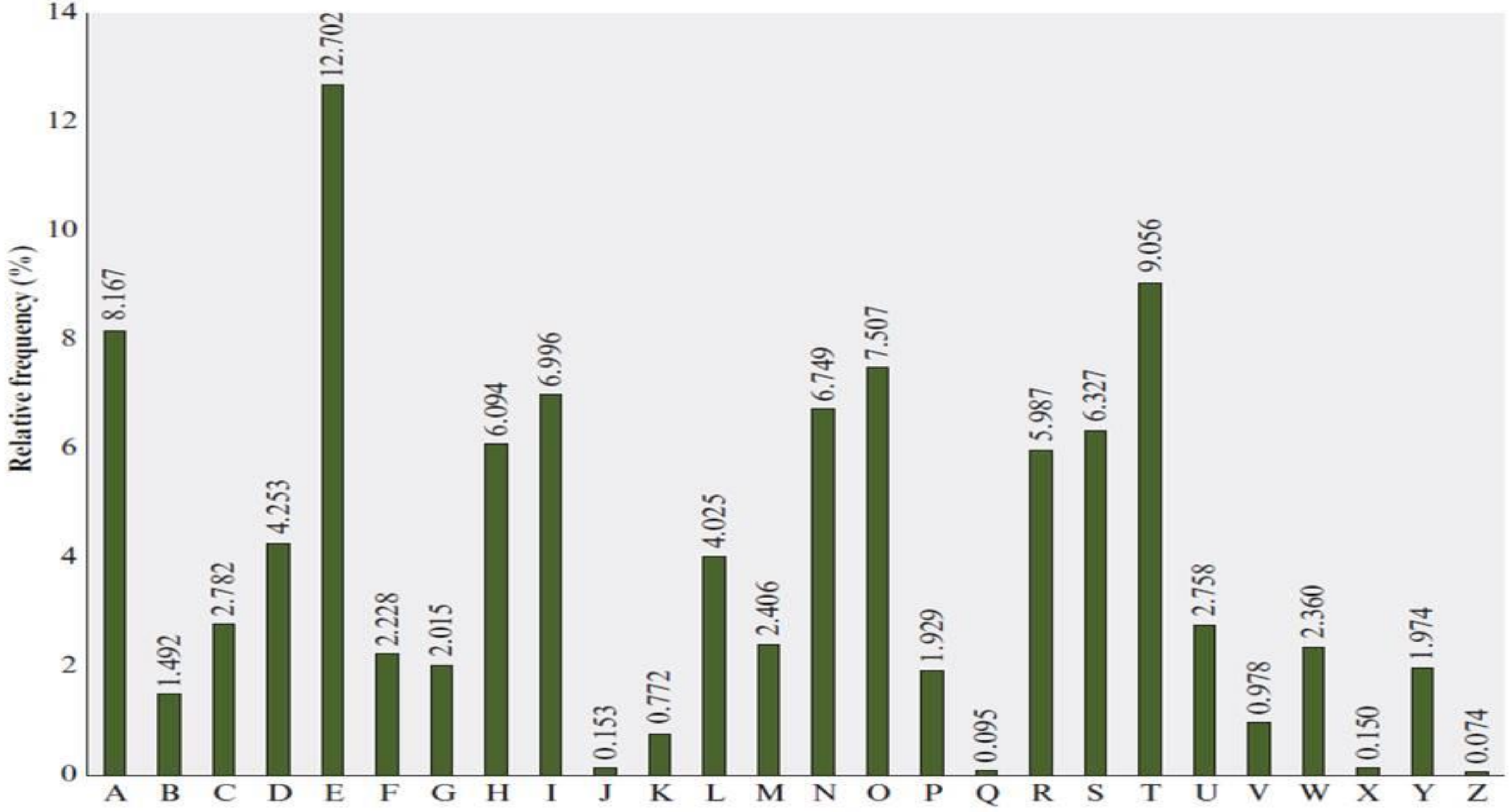
MEET ME AT OUR SPOT

↓

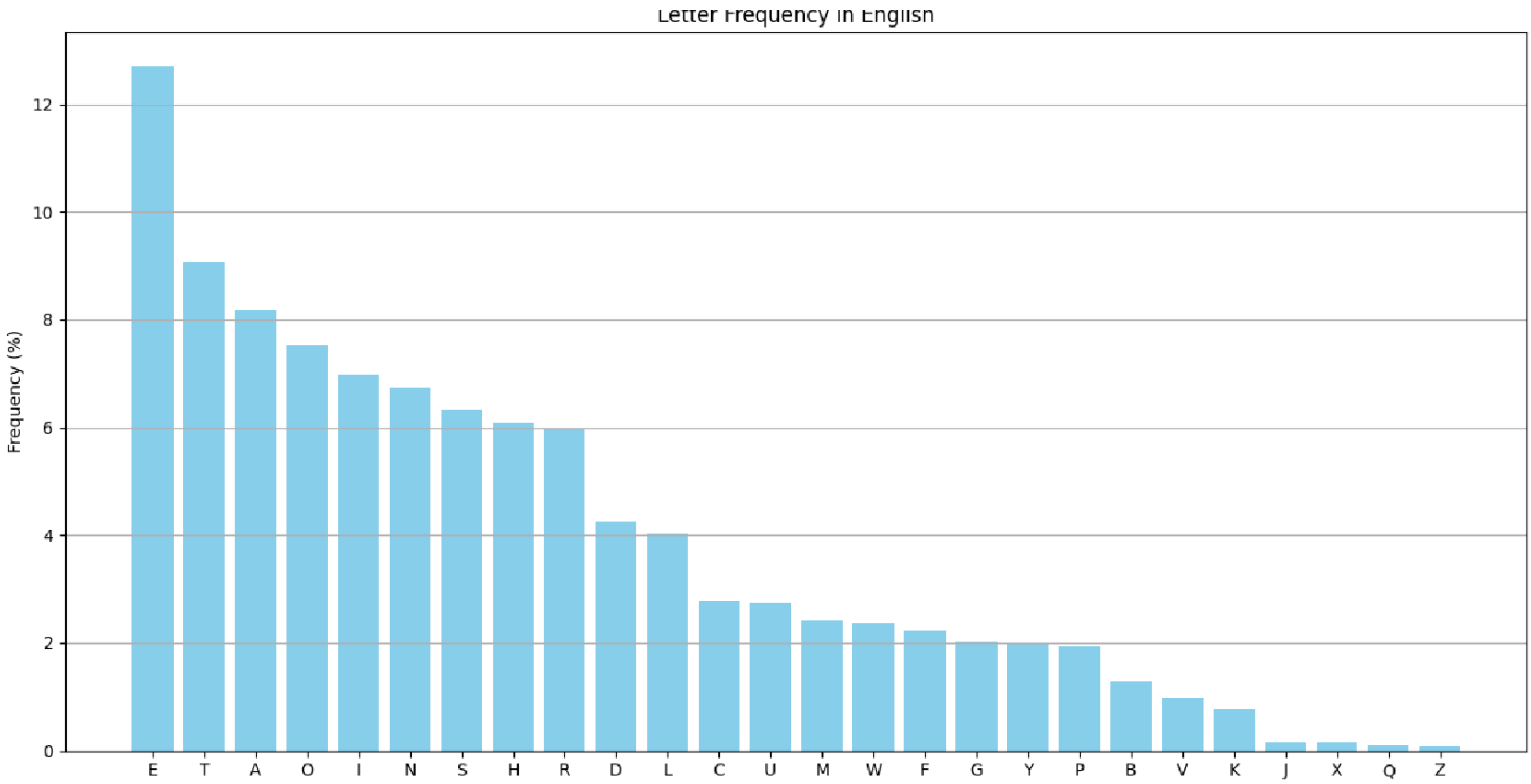
DTTM DT AM GWK LHGM

If the “cipher” line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4×10^{26} possible keys
This is 10 orders of magnitude greater than the key space for DES

Relative Frequency of Letters in English Text

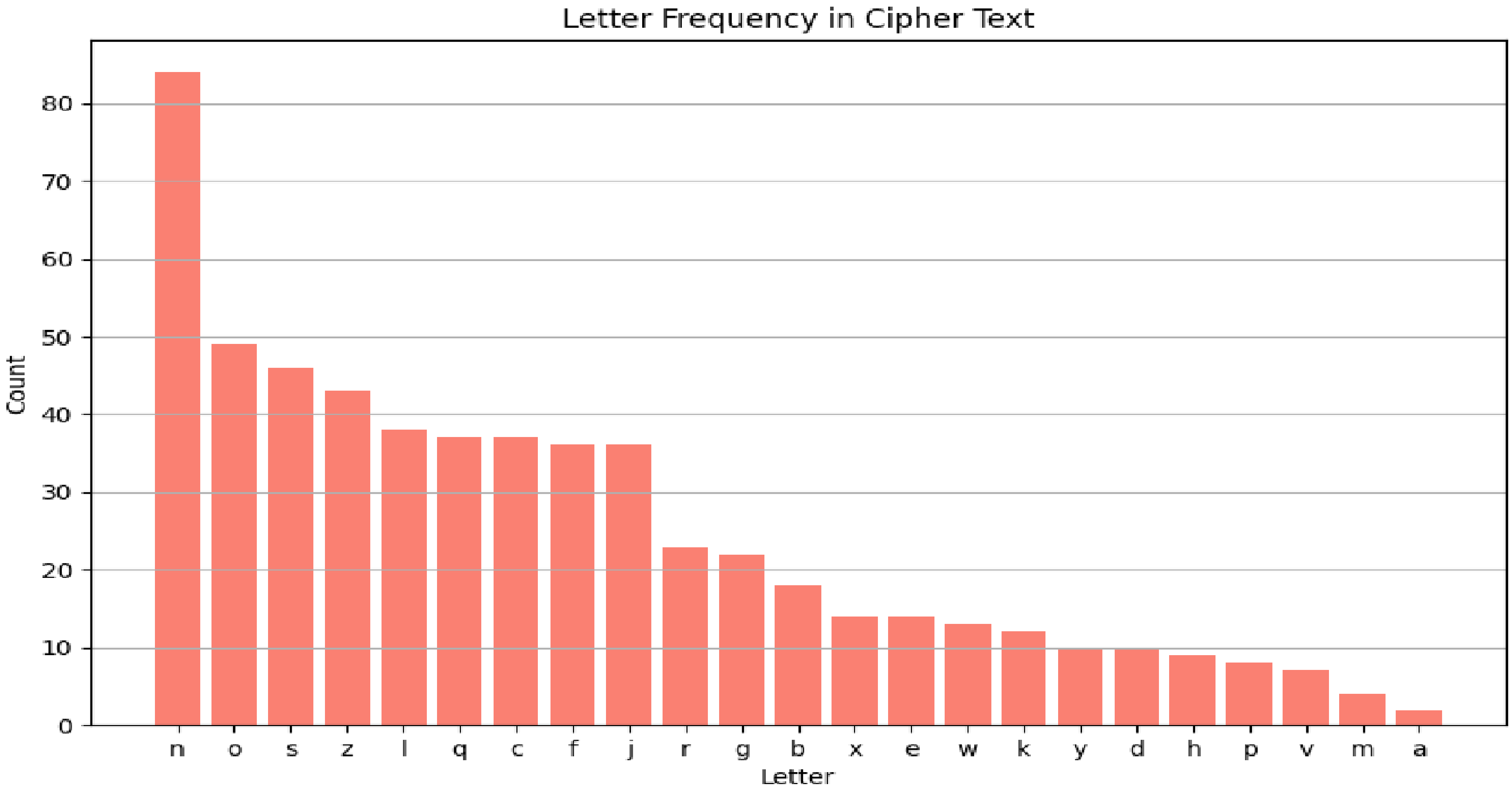


Relative Frequency of Letters in English Text



hzsrnqc klyy wqc flo mflwf ol zqdn nsoznj wskn lj xzsrbjnf, wzsxz gqv zqhhnf ol ozn glco zlfnco hnlhrn; nsoznj jnrqosdnc lj fnqj kjsnfbc, wzsxz sc xnjoqsfrv gljn efeceqr. zn rsdnb qrlfn sf zsc zlecn sf cqdsrrn jlw, wzsoznj flfn hnfnojqonb. q csfyrn blgncosx cekksxnb ol cnjdn zsg. zn pjnqmkqconb qfb bsfnb qo ozn xrep, qo zlejc gqozngqosxqrrv ksanb, sf ozn cqgn jllg, qo ozn cqgn oqprn, fndnj oqmsfy zsc gnqrc wsoz loznj gngpnjc, gexz rncc pjsfysfy q yenco wsoz zsg; qfb wnfo zlgn qo naqxorv gsbfsyzo, lfrv ol jnosjn qo lfxn ol pnb. zn fndnj ecnb ozn xlcv xzqgpnjc wzsxz ozn jnkljg hjldsbnc klj soc kqdlejnb gngpnjc. zn hqccnb onf zlejc leo lk ozn ownfov-klej sf cqdsrrn jlw, nsoznj sf crnnhsfy lj gqmsfy zsc olsrno.

Cryptanalysis on monoalphabetic cipher



Polyalphabetic Cipher

- Polyalphabetic Cipher is a substitution cipher in which the cipher alphabet for the plain alphabet may be different at different places during the encryption process;
  - Playfair Cipher: replace 2 characters by 2 characters
  - Hill Cipher: replace 3 characters by 3 characters
  - Vigenere Cipher

Playfair encryption

+1

Key matrix

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

+1

Plaintext: "Hide the gold in the tree stump"

Plaintext diagram:

HI DE TH EG OL DI NT HE TR EX ES TU MP

Ciphertext diagram:

BF CK PD FI .. .. ZD

https://en.wikipedia.org/wiki/Playfair_cipher

(4) Hill Cipher

0	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
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

- Developed by the mathematician Lester Hill in 1929
- Strength is that it completely hides single-letter frequencies
  - The use of a larger matrix hides more frequency information
  - A 3 x 3 Hill cipher hides not only single-letter but also two-letter frequency information
- Strong against a ciphertext-only attack but easily broken with a known plaintext attack

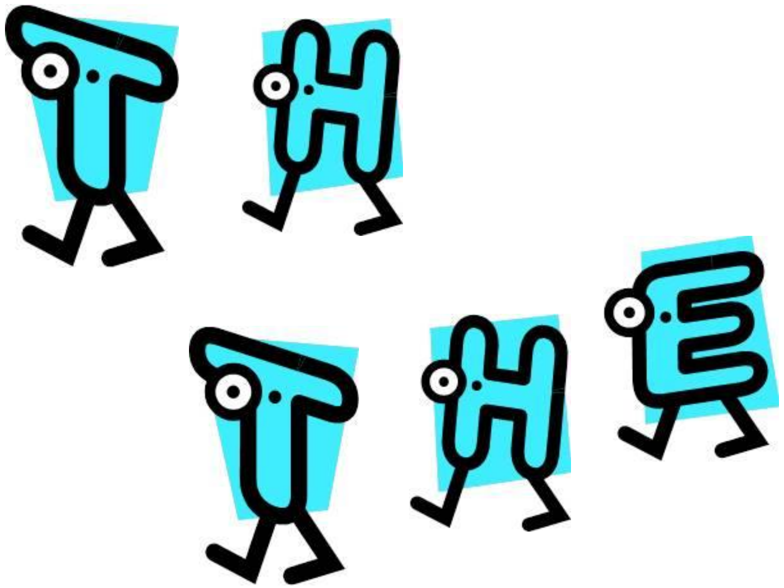
$$C = K.P \bmod 26$$

$$\begin{pmatrix} k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,1} & k_{3,2} & k_{3,3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \bmod 26$$

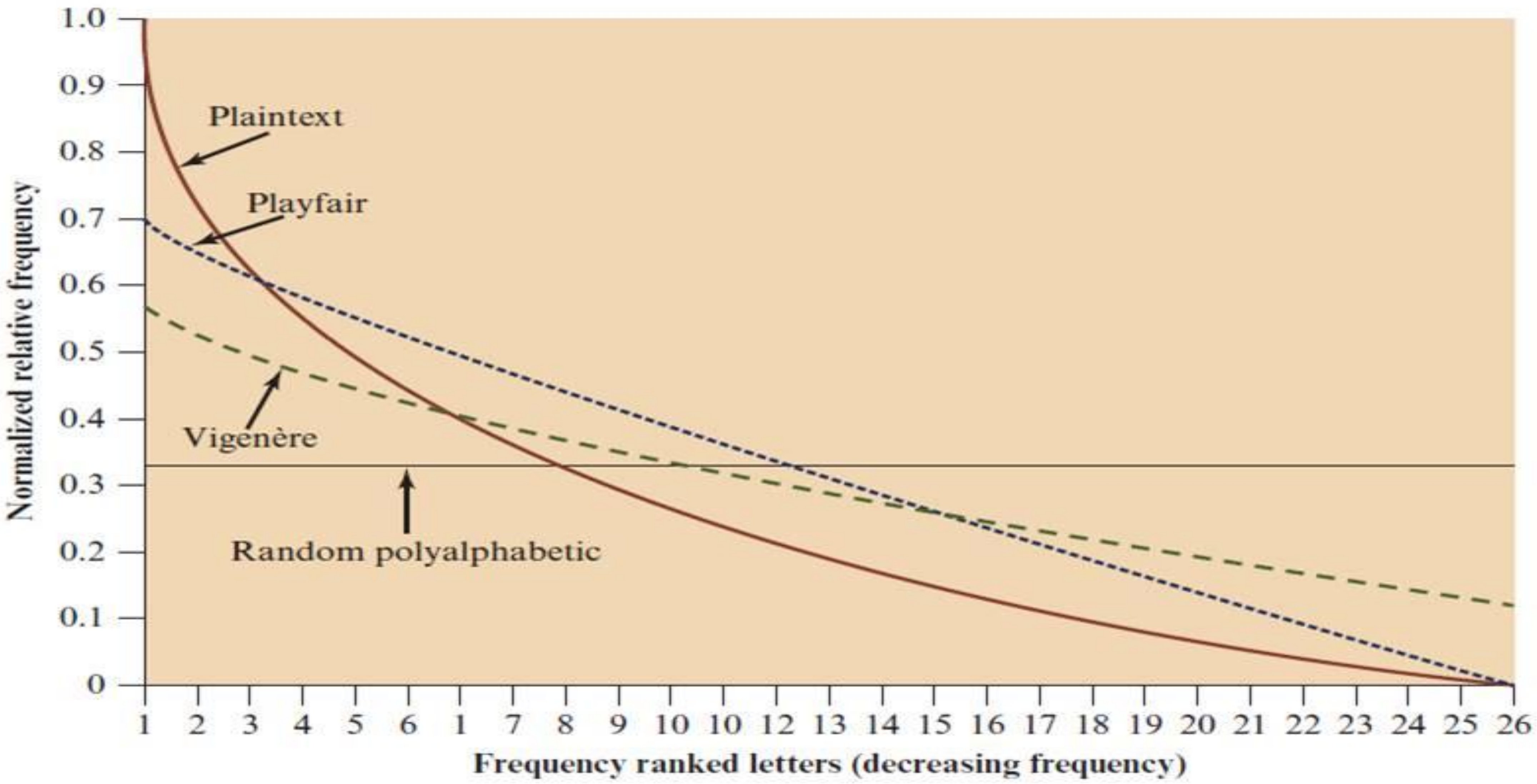
Polyalphabetic Cipher

Cryptoanalys Playfair cipher

- Digram
  - Two-letter combination
  - Most common is *th*
- Trigram
  - Three-letter combination
  - Most frequent is *the*



Classical symmetric cipher cryptanalysis



(5) Vigenère Cipher

- Best known and one of the simplest polyalphabetic substitution ciphers
- In this scheme the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers with shifts of 0 through 25
- Each cipher is denoted by a key letter which is the ciphertext letter that substitutes for the plaintext letter a

https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher

Example of Vigenère Cipher

- To encrypt a message, a key is needed that is as long as the message
- Usually, the key is a repeating keyword
- For example, if the keyword is *deceptive*, the message "we are discovered save yourself" is encrypted as:

plaintext: wearediscoveredsaveyourself

key: deceptivedeceptivedeceptive

ciphertext: ??

Vigenère Cipher

- Vigenère matrix

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

Vigenère Autokey System

- Example:

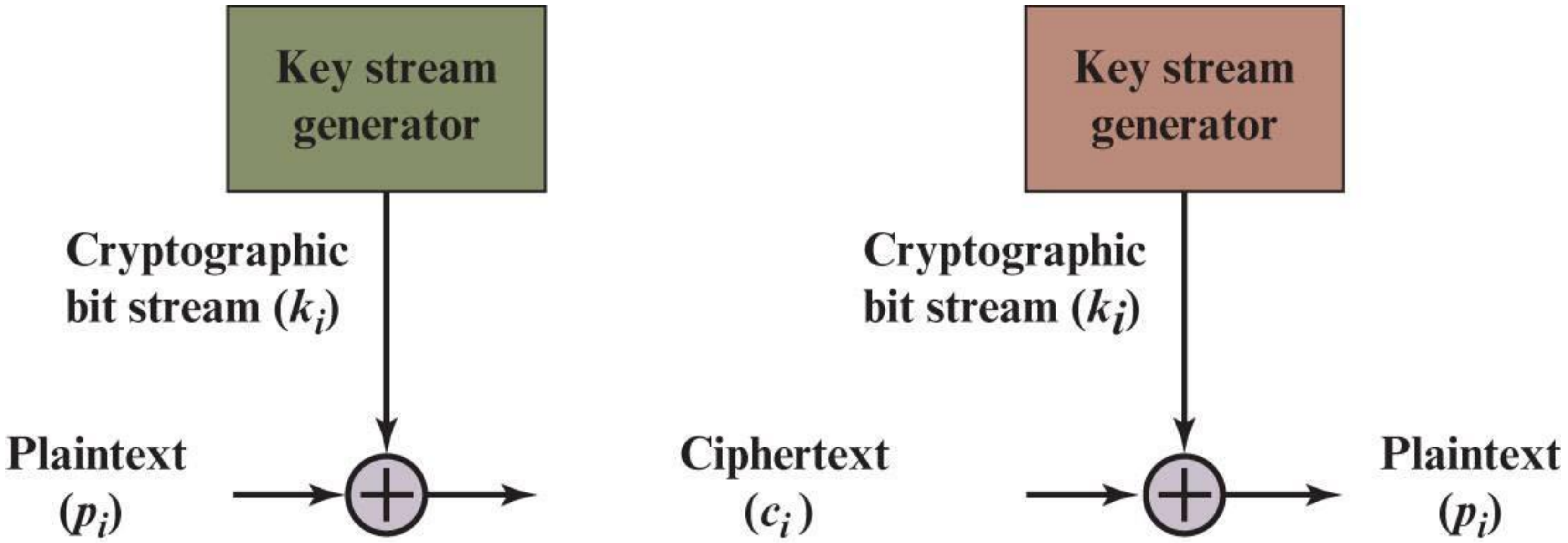
  key: deceptivewearediscoveredsav

  plaintext: wearediscoveredsaveyourself

  ciphertext: ZICVTWQNGKZEIIGASXSTSLVVWLA

- Even this scheme is vulnerable to cryptanalysis
  - Because the key and the plaintext share the same frequency distribution of letters, a statistical technique can be applied

(6) Vernam Cipher



https://en.wikipedia.org/wiki/Gilbert_Vernam

One-Time Pad

- Improvement to Vernam cipher proposed by an Army Signal Corp officer, Joseph Mauborgne
- Use a **random key that is as long as the message** so that the key need not be repeated
- Key is used to encrypt and decrypt a single message and then is discarded
- Each new message requires a new key of the same length as the new message
- **Scheme is unbreakable**
  - Produces random output that bears no statistical relationship to the plaintext
  - Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code

Difficulties

- The one-time pad offers complete security but, in practice, has two fundamental difficulties:
  - There is the practical problem of making large quantities of random keys
    - Any heavily used system might require millions of random characters on a regular basis
  - Mammoth key distribution problem
    - For every message to be sent, a key of equal length is needed by both sender and receiver
- Because of these difficulties, the one-time pad is of limited utility
  - Useful primarily for low-bandwidth channels requiring very high security
- The one-time pad is the only cryptosystem that exhibits *perfect secrecy* (see Appendix F)

Transposition ciphers

Goals: scrambles the positions of characters

(1) Rail fence cipher

(2) Columnar Transposition Cipher

https://en.wikipedia.org/wiki/Transposition_cipher

Transposition cipher

(1) Rail fence cipher

- Simplest transposition cipher
- Plaintext is written down as a sequence of diagonals and then read off as a sequence of rows
- To encipher the message “meet me after the toga party” with a rail fence of depth 2, we would write:

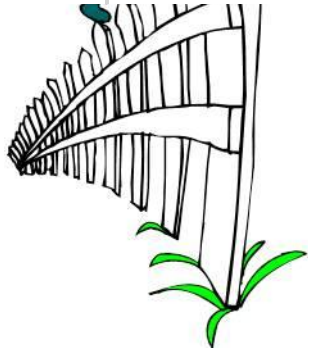
m	e	m	a	t	r	h	t	g	p	r	y
	e	t	e	f	e	t	e	o	a	a	t

Ciphertext

Encrypted message is:

MEMATRHTGPRYETEFETEOAAT

https://en.wikipedia.org/wiki/Rail_fence_cipher



Columnar Transposition Cipher

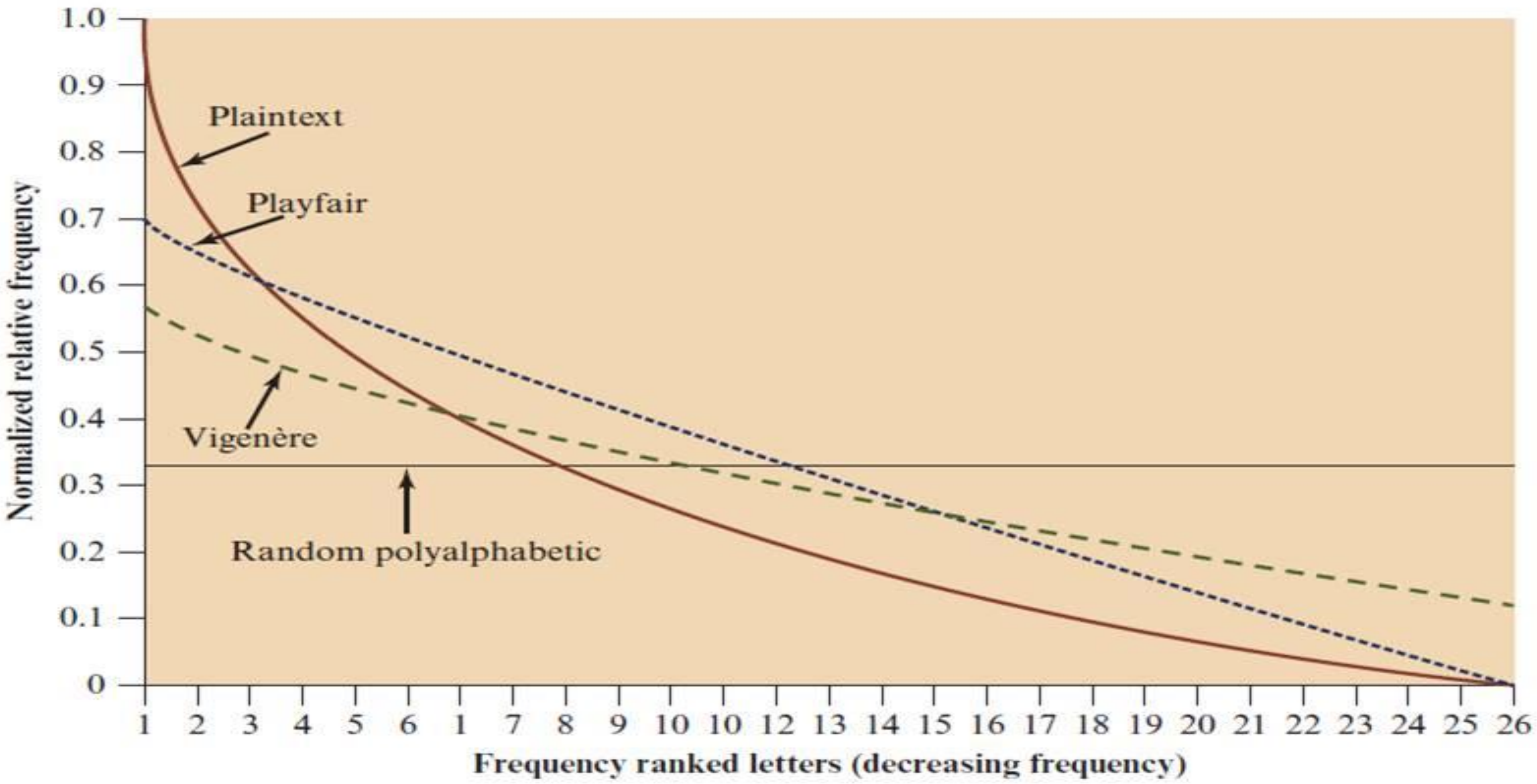
- Is a more complex transposition
- Write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns
  - ➢ The order of the columns then becomes the key to the algorithm

Key:	4	3	1	2	5	6	7
Plaintext	a	t	t	a	c	k	p
	o	s	t	p	o	n	e
	d	u	n	t	i	l	t
	w	o	a	m	x	y	z

Ciphertext

Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

Classical symmetric cipher cryptanalysis



- **Secret key (Keystream)**

$$K = k_1 k_2 \cdots k_i \cdots$$

- **Plaintext stream**

$$M = m_1 m_2 \cdots m_i \cdots$$

m_i : bit or byte

➢ **Ciphertext**

$$C = c_1 c_2 \cdots c_i \cdots$$

where $c_i = m_i \overline{\oplus} k_i$

$k1$	$k2$	$k3$	...	k_n
$m1$	$m2$	$m3$	...	m_n

$k1 \oplus m1$ $\quad$ $k2 \oplus m2$ $\quad$ ... $\quad$ $k_n \oplus m_n$

Stream Cipher (2 of 8)

Vigenère cipher

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
0	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

- **Plaintext stream**

M =	A	T	T	A	C	K	A	T	D	A	W	N
	0	19	19	0	2	10	0	19	3	0	22	13

- **Secret key (Keystream)**

K' =	L	E	M	O	N	L	E	M	O	N	L	E
	11	4	12	14	13	11	4	12	14	13	11	4

- **Ciphertext**

C =	L	X	F	O	P	V	E	F	R	N	H	R
	11	23	5	14	15	21	4	5	17	13	7	17

$C = c_1 c_2 \cdots c_i \cdots$ where $c_i = m_i + k_i \bmod 26$

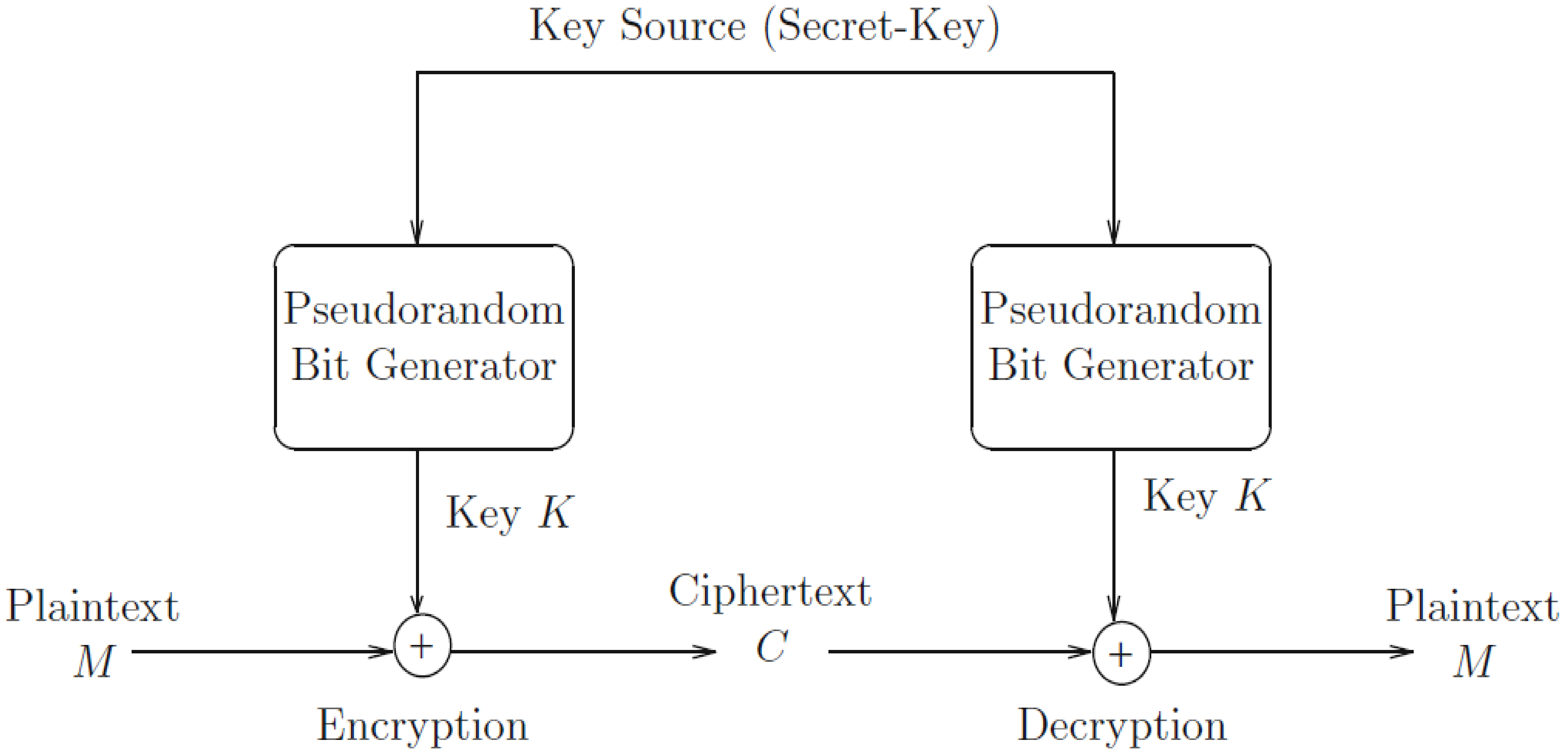
Stream Cipher (3 of 8)

- Encrypts a digital data stream **one bit or one byte** at a time
  - Examples:
    - **Autokeyed** Vigenère cipher
    - Vernam cipher
- In the ideal case, a one-time pad version of the Vernam cipher would be used, in which the keystream is as long as the plaintext bit stream
  - If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the keystream
    - Keystream must be provided to both users in advance via some independent and secure channel
    - This introduces insurmountable logistical problems if the intended data traffic is very large

Stream Cipher (4 of 8)

- For practical: must be implemented as an algorithmic to **generate key bit stream** (both users)
  - It must be computationally impractical to predict future portions of the bit stream based on previous portions of the bit stream
  - The two users need only share the generating key and each can produce the keystream

Stream Cipher (5 of 8)



Stream Cipher (6 of 8)

- **Rivest Cipher 4**

https://en.wikipedia.org/wiki/RC4

- **Chaotic-based cryptosystem**

https://en.wikipedia.org/wiki/List_of_chaotic_maps

V · T · E	Stream ciphers	
Widely used ciphers	A5/1 · A5/2 · ChaCha · Crypto-1 · E0 · **RC4**	
eSTREAM Portfolio	**Software**	HC-256 · Rabbit · Salsa20 · SOSEMANUK
	Hardware	Grain · MICKEY · Trivium
Other ciphers	Achterbahn · F-FCSR · FISH · ISAAC · MUGI · ORYX · Panama · Phelix · Pike · Py · QUAD · Scream · SEAL · SNOW · SOBER · SOBER-128 · VEST · VMPC · WAKE	
Generators	shrinking generator · self-shrinking generator · alternating step generator	
Theory	block ciphers in stream mode · shift register · LFSR · NLFSR · T-function · IV	
Attacks	correlation attack · correlation immunity · stream cipher attacks	

➢ **Chaotic-based cryptosystem**

Example:

Logistic map

$$x_{n+1} = rx_n(1 - x_n)$$

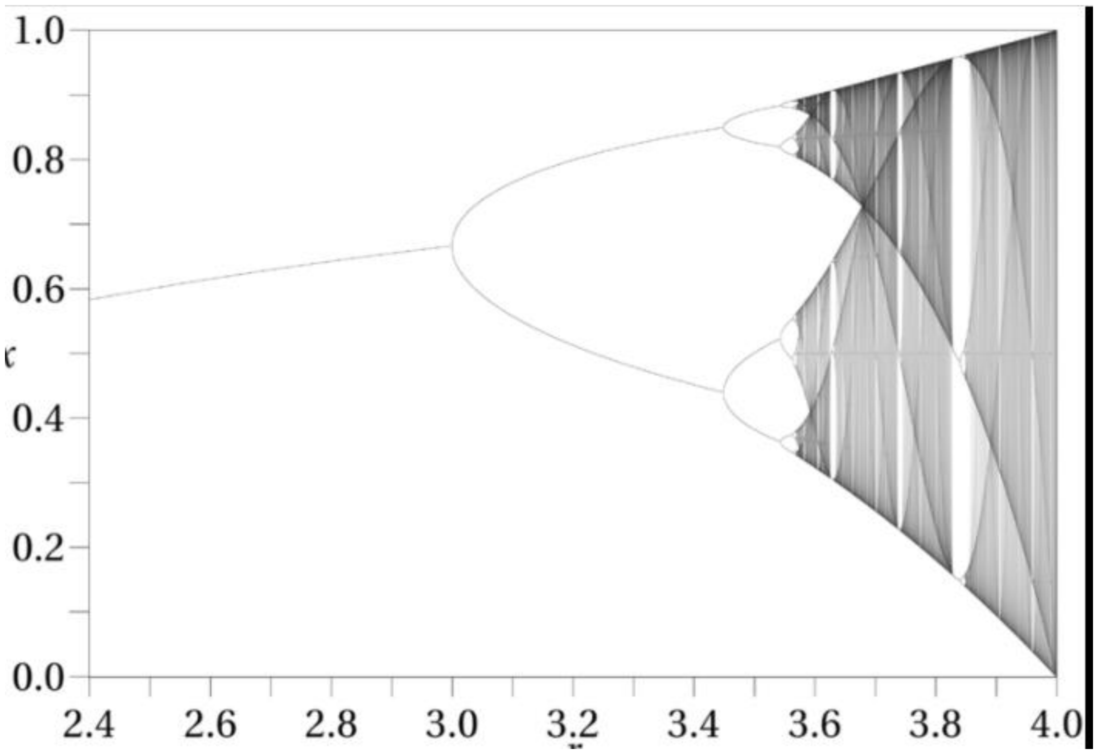
Input:

$x_0, r \in (3.6, 4)$

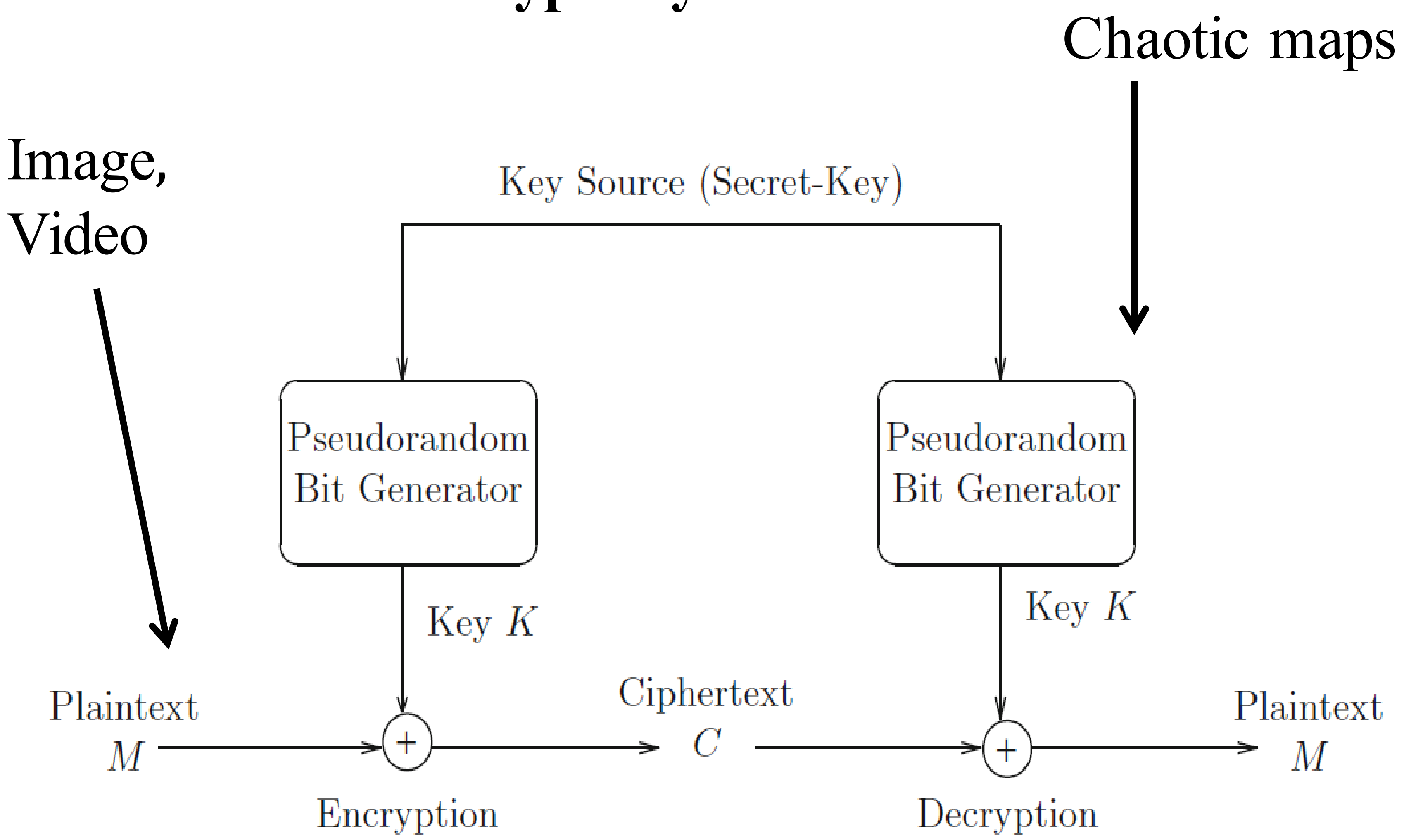
Output:

$x_1, x_2, x_3, \ldots x_n, \ldots .$

$$0 < x_i < 1$$



- **Chaotic-based crypto system**



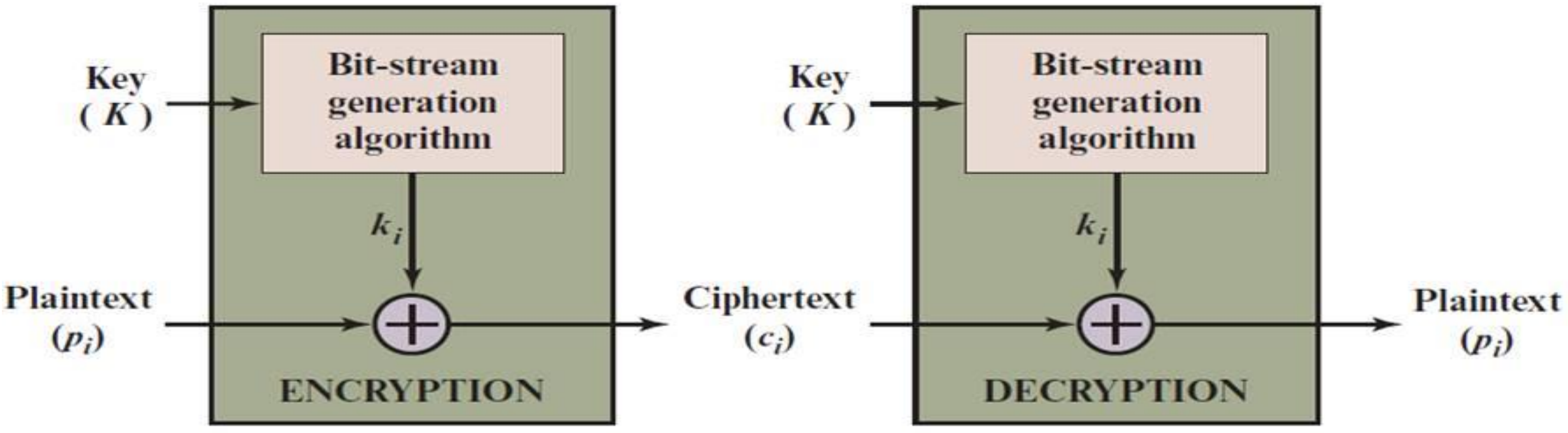
Outline

- Stream Cipher
- Block cipher
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard (AES)
  - Some other ciphers
    - Searchable encryption

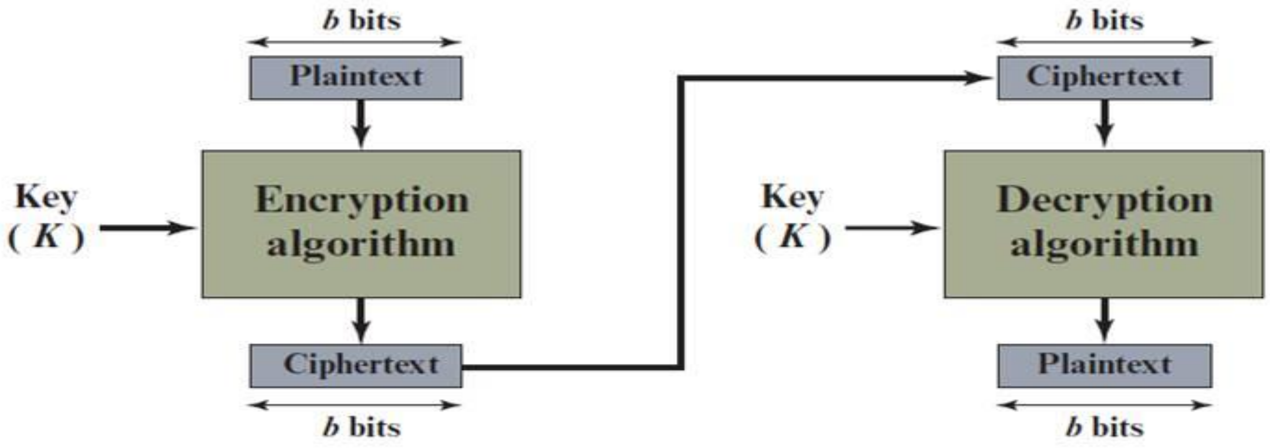
Block Cipher

- **A *block of plaintext* is treated as a whole** and used to produce a *ciphertext block of equal length*
- Typically a block size of 64 or 128 bits is used
- As with a stream cipher, the two users share a symmetric encryption key
- **The majority of network-based symmetric cryptographic applications make use of block ciphers**

Stream Cipher Vs. Block Cipher



(a) Stream cipher using algorithmic bit-stream generator



(b) Block cipher

Encryption and Decryption Tables for Substitution Cipher

$b_1 b_2 b_3 b_4$

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

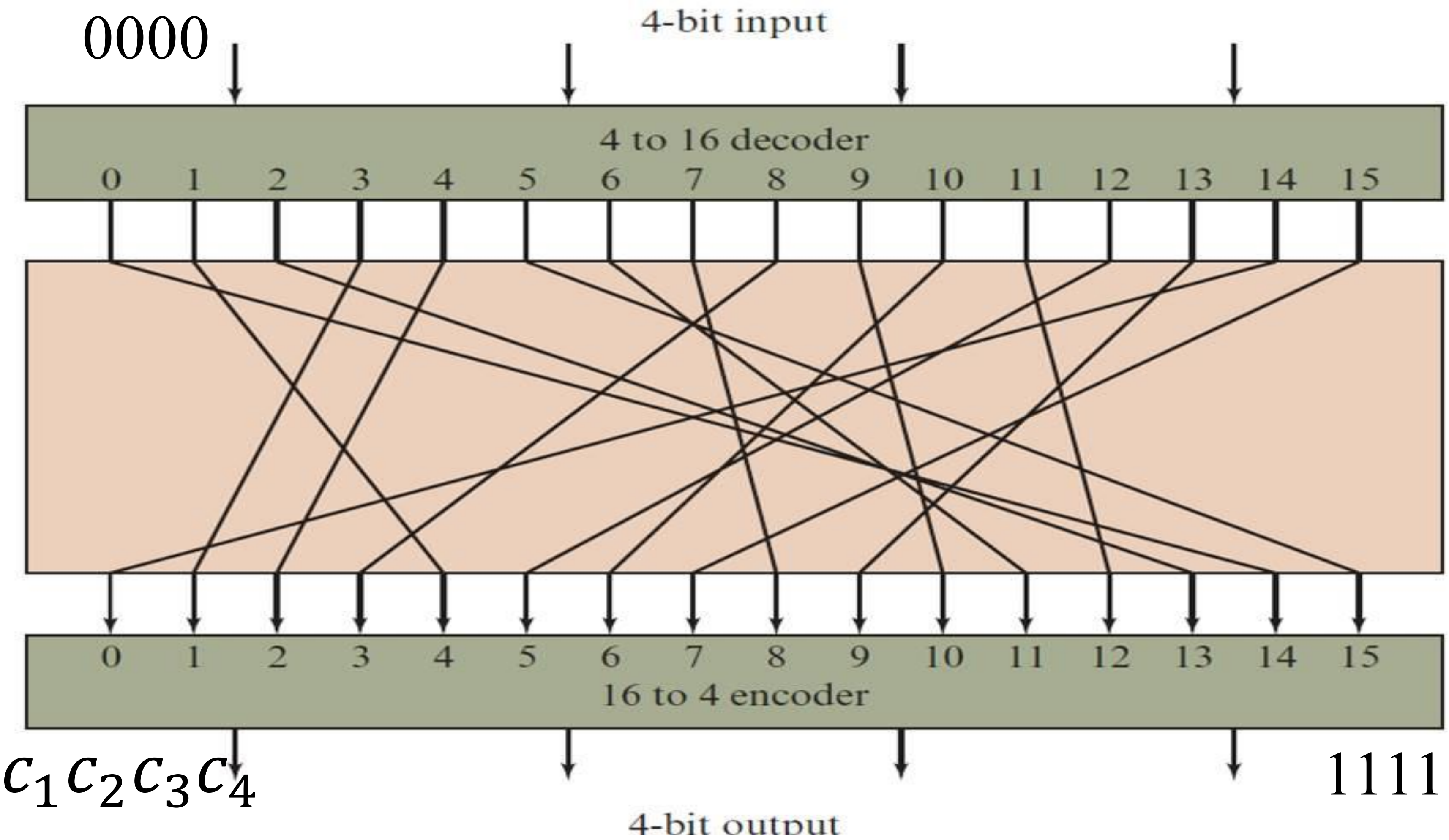
Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

Block Substitution

4-bits block substitution

$b_1 b_2 b_3 b_4$

0000



$c_1 c_2 c_3 c_4$

1111

Block Substitution

How many possible substitutions for a block n-bit?

$b_1 b_2 b_3 \ldots . b_n$

$c_1 c_2 c_3 \ldots . c_n$

Feistel Cipher

- Feistel proposed the use of a cipher that alternates substitutions and permutations
- **Substitutions**
  - Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements
- **Permutation**
  - No elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed
- Is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions
- Is the structure used by many significant symmetric block ciphers currently in use

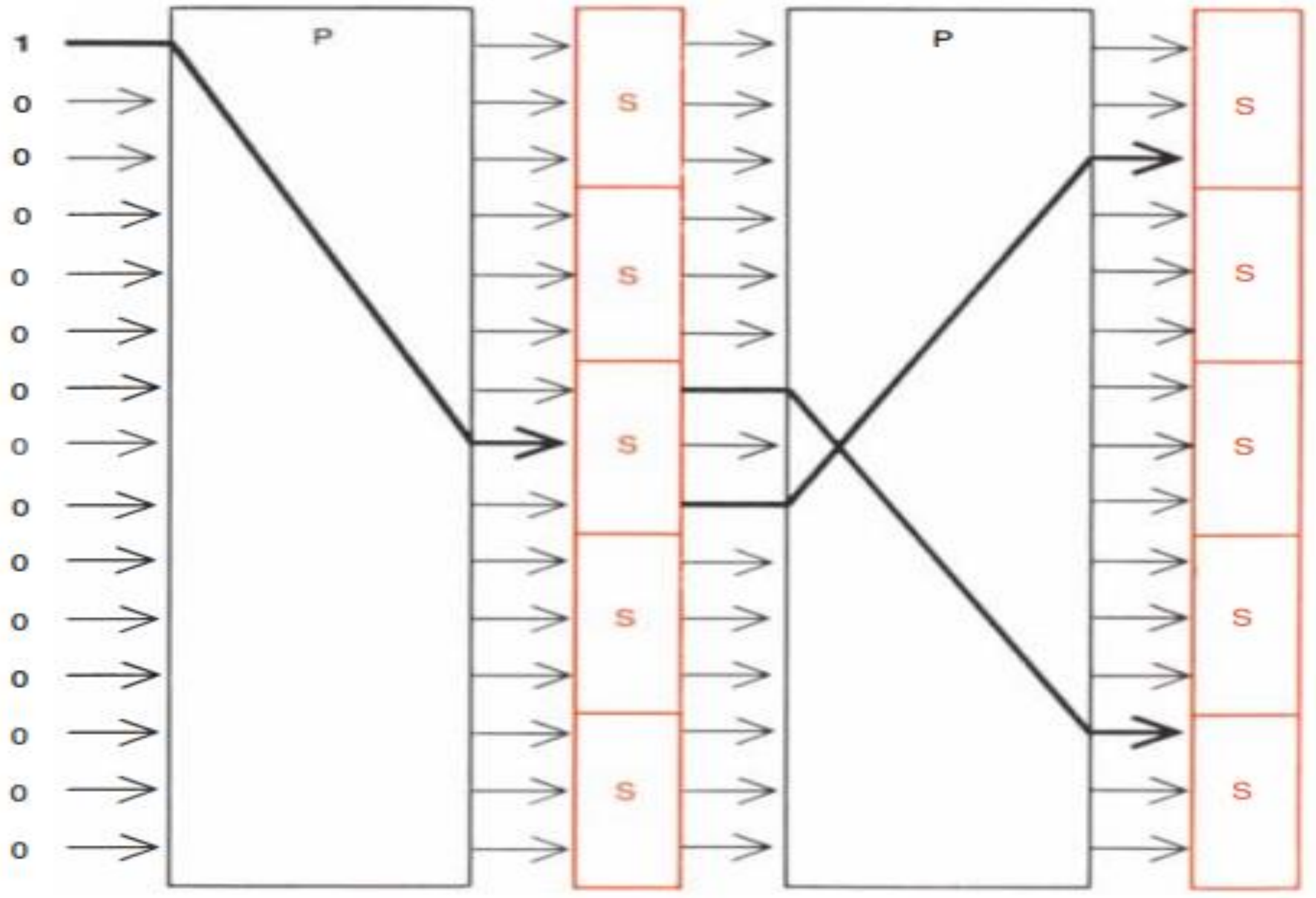
Feistel, H. (1973). Cryptography and computer privacy. Scientific american, 228(5), 15-23.

Diffusion and Confusion

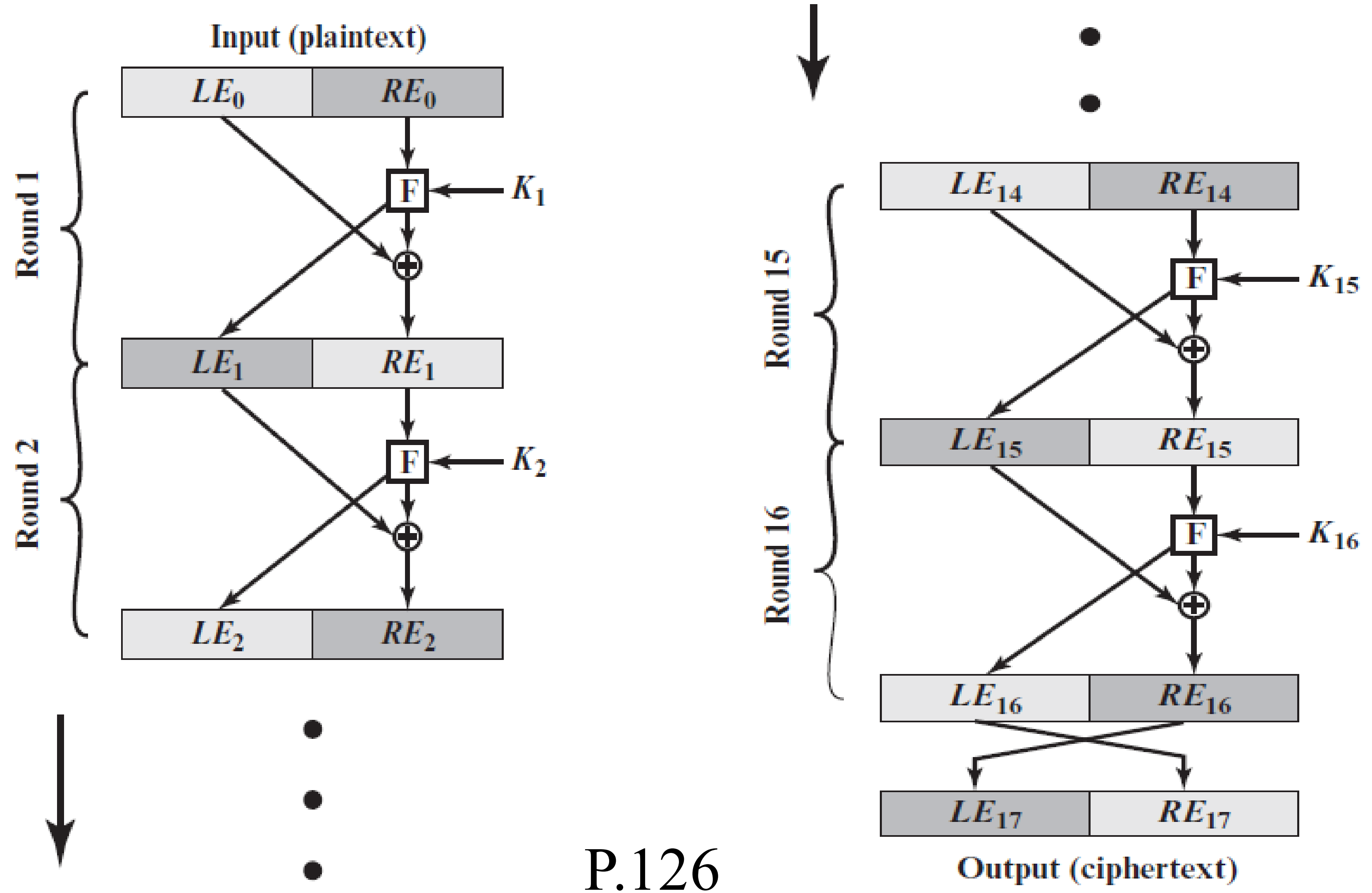
- Terms introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system
  - Shannon’s concern was to thwart cryptanalysis based on statistical analysis

- **Diffusion**
  - The statistical structure of the **plaintext** is dissipated into long-range statistics of the **ciphertext**
  - This is achieved by having each plaintext digit affect the value of many ciphertext digits
- **Confusion**
  - Seeks to make the relationship between the statistics of the **ciphertext** and the value of the **encryption key** as complex as possible
  - Even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key

Feistel Cipher

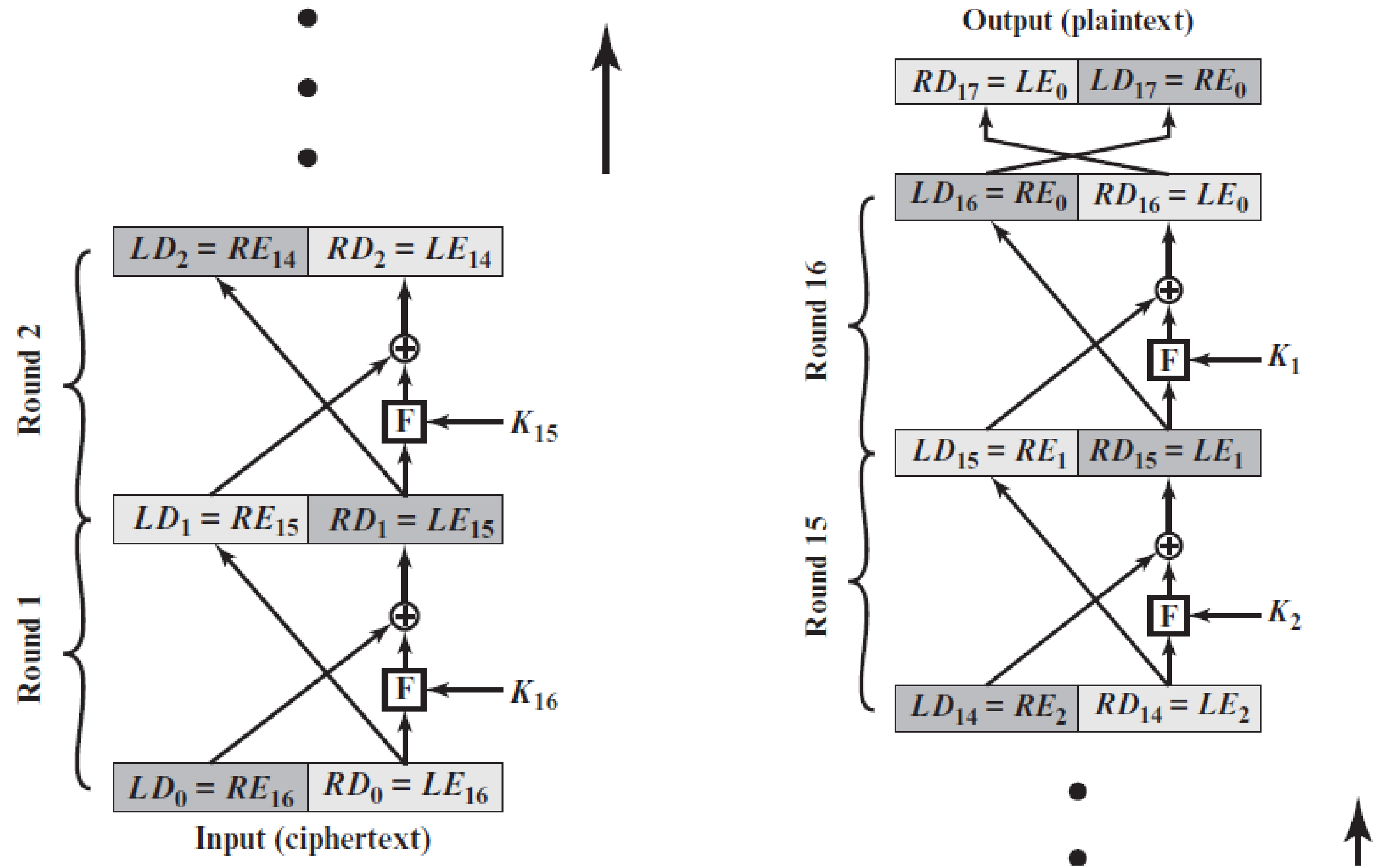


Feistel Encryption and Decryption (16 rounds)



P.126

Feistel Encryption and Decryption (16 rounds)



The Feistel Cipher Scheme (FCS)

- Divide M into blocks of $2l$-bits long (pad the last block if needed)
- Use only the XOR and Substitution operations
- Generate *n* sub-keys of a fixed length from the encryption key K: $K_1,\ldots,K_n$
- Divide a $2l$-bit block input into two parts: L_0 and R_0, both of size l (the suffix and prefix of the block, respectively)
- Perform a substitution function F on an l-bit input string and a sub-key to produce an l-bit output
- Encryption and decryption each executes n rounds of the same sequence of operations

FCS Encryption and Decryption

FCS Encryption

- Let $M = L_0R_0$; execute the following operations in round i, $i = 1, \ldots, n$:

  $L_i = R_{i-1}$

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

- Let $L_{n+1} = R_n$, $R_{n+1} = L_n$ and $C = L_{n+1}R_{n+1}$

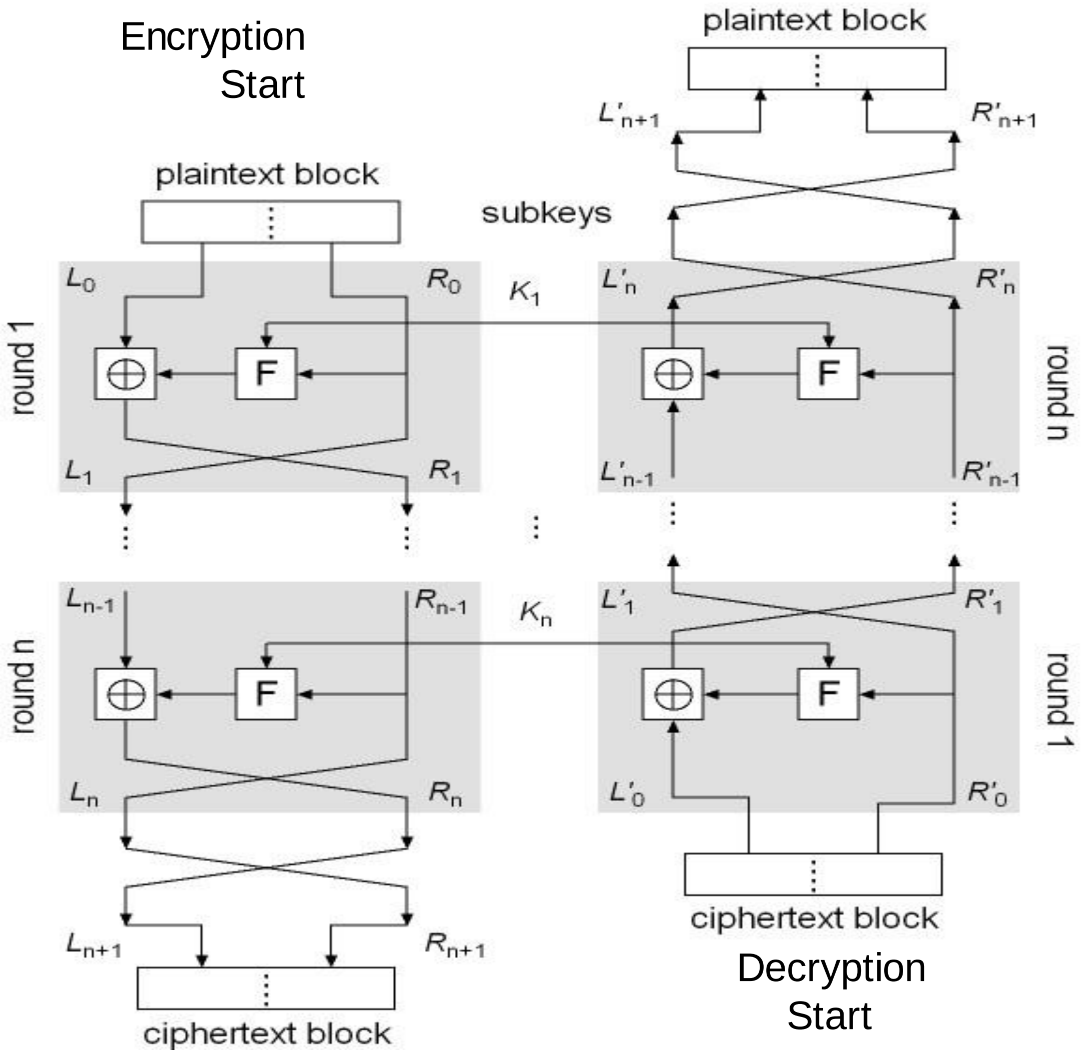
FCS Decryption

- Symmetrical to encryption, with sub-keys in reverse order
- Rewrite C as $C = L'_0R'_0$
- Execute the following in round i ($i = 1, \ldots, n$):

  $L'_i = R'_{i-1}$

  $R'_i = L'_{i-1} \oplus F(R'_{i-1}, K'_{n-i+1})$

- Let $L'_{n+1} = R'_n$, $R'_{n+1} = L'_n$
- We will show that $M = L'_{n+1}R'_{n+1}$



Proof of FCS decryption

- Will show that $C = L_{n+1}R_{n+1} = L'_0R'_0$ is transformed back to $M = L_0R_0$ by the FCS Decryption algorithm
- Prove by induction the following equalities:

  (1) $L'_i = R_{n-i}$ (2) $R'_i = L_{n-i}$
- **Basis**: $L_0' = L_{n+1} = R_n$, $R_0' = R_{n+1} = L_n$; (1) and (2) hold
- **Hypothesis**: Assume when $i \le n$:

  $L_{i-1}' = R_{n-(i-1)}$ $R_{i-1}' = L_{n-(i-1)}$
- **Induction step**:

  $L'_i = R'_{i-1}$ (by decrypt. alg.) $= L_{n-i+1}$ (by hypothesis) $= R_{n-i}$ (by encrypt. alg.)

  Hence (1) is true
- $R'_i = L'_{i-1} \oplus F(R'_{i-1}, K_{n-i+1})$

  $= R_{n-(i+1)} \oplus F(L_{n-(i+1)}, K_{n-i+1})$

  $= [L_{n-i} \oplus F(R_{n-i}, K_{n-i+1})] \oplus F(R_{n-i}, K_{n-i+1})$

  $= L_{n-i}$

  Hence (2) true

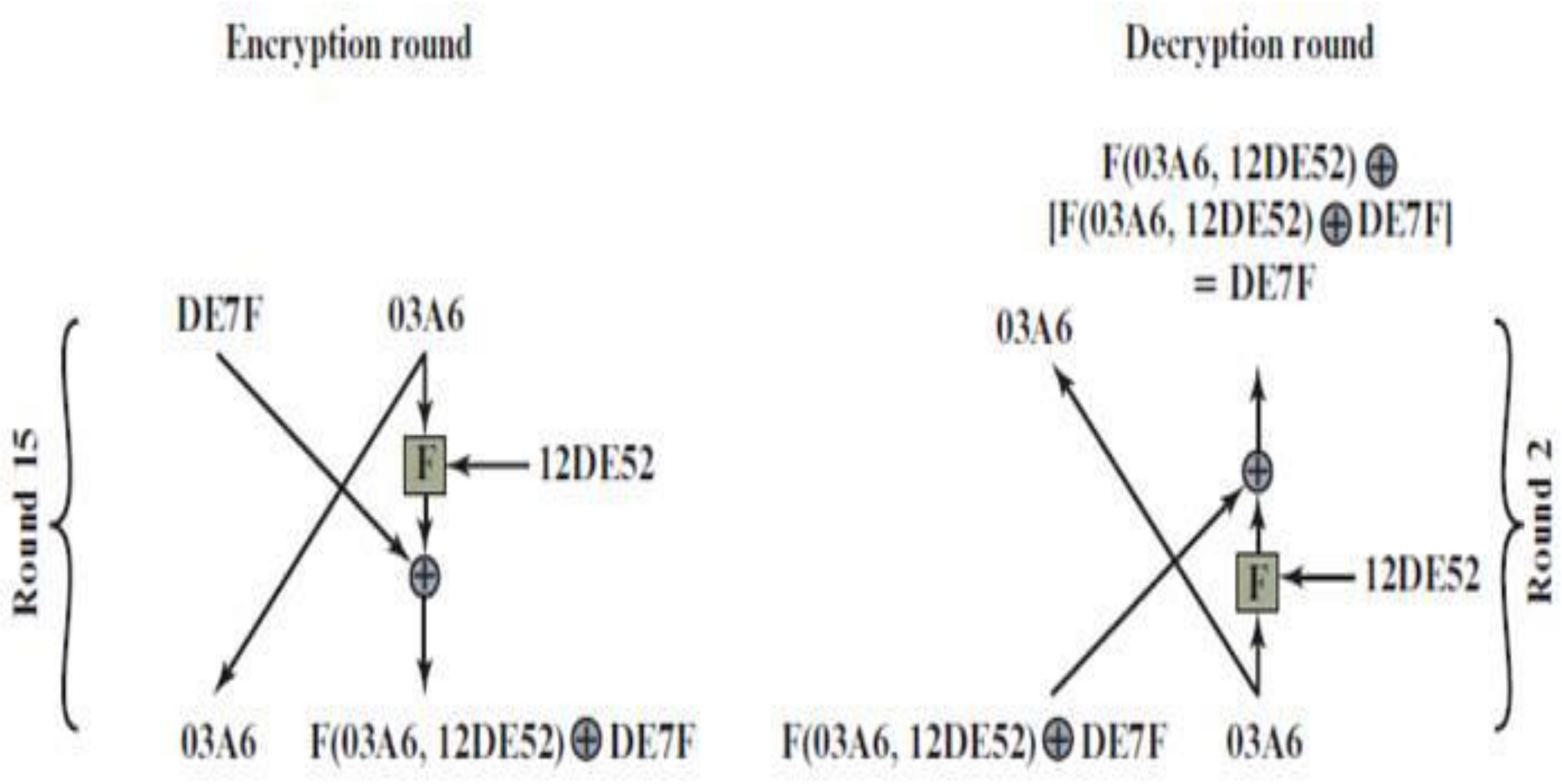
Feistel Cipher Design Features (1 of 2)

- Block size
  - Larger block sizes mean greater security but reduced encryption/decryption speed for a given algorithm
- Key size
  - Larger key size means greater security but may decrease encryption/decryption speeds
- Number of rounds
  - The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security
- Subkey generation algorithm
  - Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis

Feistel Cipher Design Features (2 of 2)

- Round function F
  - Greater complexity generally means greater resistance to cryptanalysis
- Fast software encryption/decryption
  - In many cases, encrypting is embedded in applications or utility functions in such a way as to preclude a hardware implementation; accordingly, the speed of execution of the algorithm becomes a concern
- Ease of analysis
  - If the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength

Feistel Example



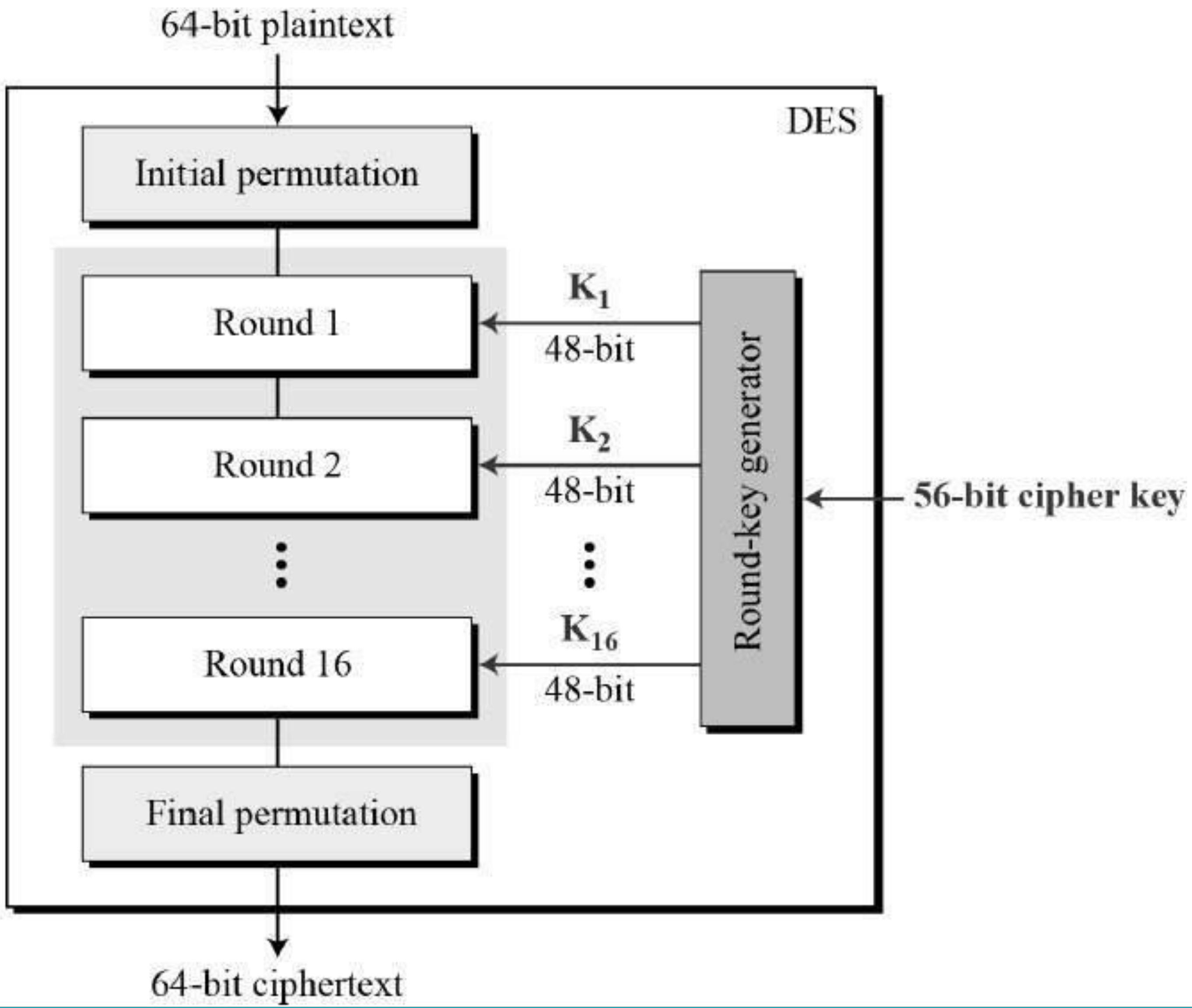
Outline

- Stream Cipher
- Block cipher
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard (AES)
  - Some other ciphers
    - Searchable encryption

Data Encryption Standard (DES)

- Issued in 1977 by the National Bureau of Standards (now NIST) as Federal Information Processing Standard 46
- Was the most widely used encryption scheme until the introduction of the Advanced Encryption Standard (AES) in 2001
- Algorithm itself is referred to as the Data Encryption Algorithm (DEA)
  - Data are encrypted in 64-bit blocks using a 56-bit key
  - The algorithm transforms 64-bit input in a series of steps into a 64-bit output
  - The same steps, with the same key, are used to reverse the encryption

DES Encryption Algorithm



DES Sub-Key Generation

- The block size of DES is 64 bits and the encryption key is 56 bits, which is represented as a 64-bit string $K = k_1\ k_2\ \ldots\ k_{64}$
- DES uses 16 rounds of iterations with 16 sub-keys
- Sub-key generation:
  1. Remove the $8i$-th bit (i = 1, 2, …, 8) from K
  2. Perform an ***initial permutation*** on the remaining 56 bits of K, denoted by $\mathrm{IP}_{key}(K)$
  3. Split this 56-bit key into two pieces: U_0V_0, both with 28 bits
  4. Perform Left Circular Shift on U_0 and V_0 a defined number of times, producing U_iV_i:

     $$U_i = LS_{z(i)}\ (U_{i-1}), \qquad V_i = LS_{z(i)}\ (V_{i-1})$$

  5. Permute the resulting U_iV_i using a defined compress permutation, resulting in a 48-bit string as a sub-key, denoted by K_i

     $$K_i = P_{key}\ (U_i\ V_i\ )$$

https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/

DES Substitution Boxes

- The DES substitution function F is defined below:

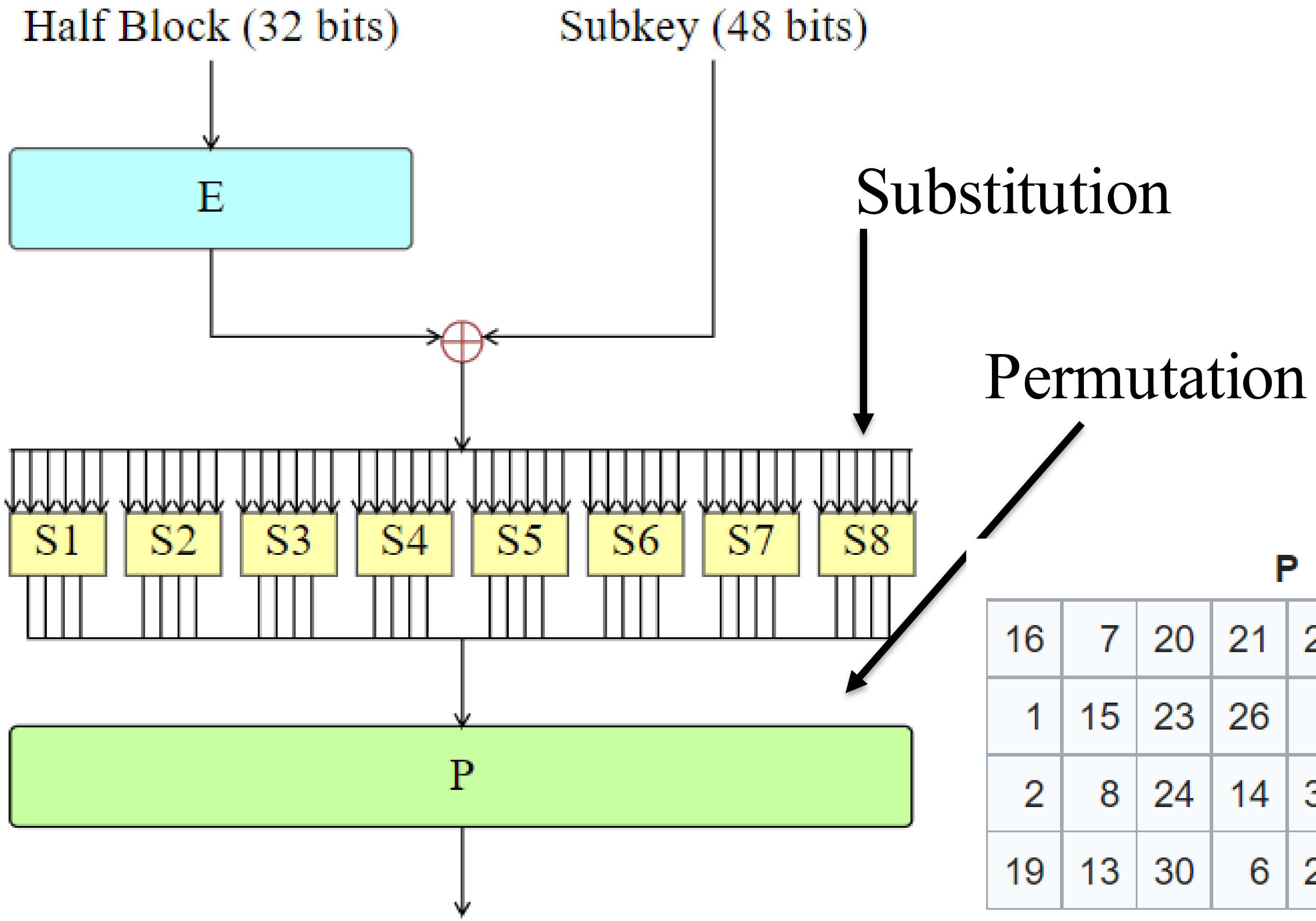
$$F(R_{i-1}, K_i) = P(S(EP(R_{i-1}) \oplus K_i)),\ i = 1,\ldots,16$$

- First, permute R_i using $EP(R_i)$ to produce a 48-bit string x
- Next, XOR x with the 48-bit sub key K_i to produce a 48-bit string y
- Function S turns y into a 32-bits string z, using eight 4x16 special matrices, called S-boxes
  - Each entry in an S-box is a 4-bit string
  - Break y into 8 blocks, each with 6-bits
  - Use the i^{th} matrix on the i^{th} block $b_1b_2b_3b_4b_5b_6$
  - Let b_1b_6 be the row number, and $b_2b_3b_4b_5$ the column number, and return the corresponding entry
  - Each 6-bit block is turned to a 4-bit string, resulting in a 32-bit string z
- Finally, permute z using P to produce the result of DES's F function
- This result, XOR'd with L_{i-1}, is R_i

https://en.wikipedia.org/wiki/DES_supplementary_material

DES function $F(R_{i-1}, K_i)$

$$F(R_{i-1}, K_i) = P(S(EP(R_{i-1}) \oplus K_i)), i = 1,\ldots,16$$



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

DES Substitution Boxes

S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Input: “**0**1101**1**”
Output: “1001”

DES encryption steps

- Rewrite $IP(M) = L_0R_0$, where $|L_0| = |R_0| = 32$
- For $i = 1, 2, \ldots, 16$, execute the following operations in order:

$$L_i = R_{i-1}$$

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

- Let $C = IP^{-1}(R_{16}L_{16})$.

Is DES good enough?

- Security strength of DES
  - Number of rounds
  - Length of encryption key
  - Construction of the substitute function
- DES was used up to the 1990's.
- People began to take on the DES Challenges to crack DES
- Only uses 56-bit keys = $2^{56} \sim 7.2 \times 10^{16}$ keys
- Brute-force will work with current technology
  - In 1997 on Internet in a few months
  - In 1998 on dedicated h/w (EFF) in a few days
  - In 1999 above combined in 22 hours

What to Do Next?

- Start over
- New standards begin to be looked into
- On the other hand, can we extend the use of DES?

Block Cipher Design Principles

- 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

- 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

- 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