Classical Ciphers - Substitution Ciphers

19CSE311 Computer Security

Jevitha KP

Department of CSE

Classical Ciphers

- Traditional symmetric-key ciphers is divided into two broad categories:
    - substitution ciphers
        - We replace one symbol in the ciphertext with another symbol
    - transposition ciphers
        - We reorder the position of symbols in the plaintext

Substitution Ciphers

- Replaces one symbol with another.
- Eg: Alphabets or Digits
- Substitution ciphers can be categorized as :
  - monoalphabetic ciphers
  - polyalphabetic ciphers.

Monoalphabetic Ciphers

- In monoalphabetic substitution, the **relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one**

- A character (or a symbol) in the plaintext is always changed to the same character (or symbol) in the ciphertext **regardless of its position** in the text.

- For example, if the algorithm says that letter A in the plaintext is changed to letter D, every letter A is changed to letter D.

Monoalphabetic Ciphers

- Eg1: h -> K ; e -> H; l -> O ; o ->R
- The cipher is probably monoalphabetic because both l's (els) are encrypted as O's.
- Plaintext: hello
- Ciphertext: KHOOR
- Eg 2:
- The cipher is not monoalphabetic because each l (el) is encrypted by a different character. The first l (el) is encrypted as N; the second as Z.
- Plaintext: hello
- Ciphertext: ABNZF

Additive Cipher

- The simplest monoalphabetic cipher is the **additive cipher.**

- This cipher is sometimes called a **shift cipher** because the encryption algorithm can be interpreted as “shift key characters down” and the encryption algorithm can be interpreted as “shift key character up”.

- Julius Caesar used an additive cipher to communicate with his officers, so additive ciphers are sometimes referred to as the **Caesar cipher**. Caesar used a key of 3 for his communications.

- Assume that the plaintext consists of lowercase letters (a to z), and that the ciphertext consists of uppercase letters (A to Z).

- To apply mathematical operations on the plaintext and ciphertext, we assign numerical values to each letter (lower- or uppercase)

Additive Cipher

- Each character (lowercase or uppercase) is assigned an integer in Z26.

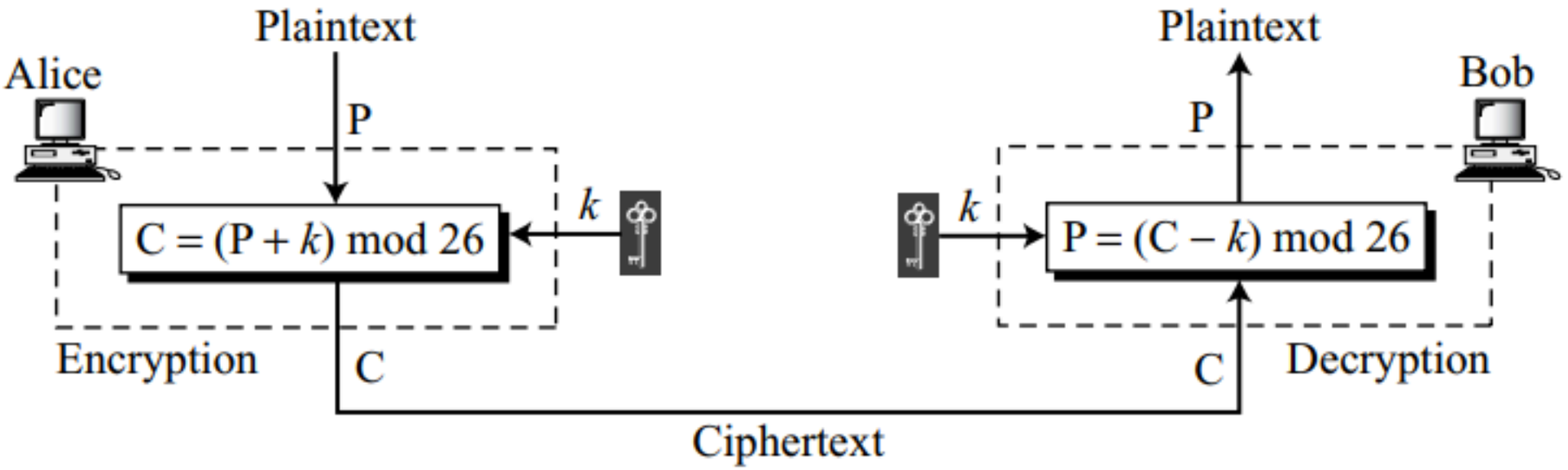
- The secret key between Alice and Bob is also an integer in Z26.

- The encryption algorithm adds the key to the plaintext character;

- the decryption algorithm subtracts the key from the ciphertext character.

- All operations are done in Z26

Additive Cipher



$$P_1 = (C - k) \bmod 26 = (P + k - k) \bmode 26 = P$$

Additive Cipher

- Use the additive cipher with key = 15 to encrypt the message “hello”.

Plaintext: h → 07	Encryption: (07 + 15) mod 26	Ciphertext: 22 → W
Plaintext: e → 04	Encryption: (04 + 15) mod 26	Ciphertext: 19 → T
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: o → 14	Encryption: (14 + 15) mod 26	Ciphertext: 03 → D

Ciphertext: W → 22	Decryption: (22 – 15) mod 26	Plaintext: 07 → h
Ciphertext: T → 19	Decryption: (19 – 15) mod 26	Plaintext: 04 → e
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: D → 03	Decryption: (03 – 15) mod 26	Plaintext: 14 → o

Cryptanalysis

- Additive ciphers are vulnerable to ciphertext-only attacks using exhaustive key searches (brute-force attacks).

- The key domain of the additive cipher is very small; there are only 26 keys.

- One of the keys, zero, is useless (the ciphertext is the same as the plaintext).

- This leaves only 25 possible keys.

- Eve can easily launch a bruteforce attack on the ciphertext

Cryptanalysis

- Ciphertext: UVACLYFZLJBYL

K = 1 → **Plaintext:** tuzbkxeykiaxk
K = 2 → **Plaintext:** styajwdxjhzwj
K = 3 → **Plaintext:** rsxzivcwigyvi
K = 4 → **Plaintext:** qrwyhubvhfxuh
K = 5 → **Plaintext:** pqvxgtaugewtg
K = 6 → **Plaintext:** opuwfsztfdvsf
K = 7 → **Plaintext:** notverysecure

Cryptanalysis

- Additive ciphers are also subject to statistical attacks.
- This is especially true if the adversary has a long ciphertext.
- The adversary can use the frequency of occurrence of characters for a particular language

Letter	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

Cryptanalysis

- Sometimes it is difficult to analyze a ciphertext based only on information about the frequency of a single letter;
- we may need to know the occurrence of specific letter combinations.

- We need to know the frequency of two-letter or three-letter strings in the ciphertext and compare them with the frequency of two-letter or three-letter strings in the underlying language of the plaintext.

Cryptanalysis

- The most common **two-letter groups (digrams)** and **three-letter groups (trigrams)** are shown below:

Digram	TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF
Trigram	THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH

Cryptanalysis

- Eve has intercepted the following ciphertext. Using a statistical attack, find the plaintext.

- XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPPEV WMXMWASVXLQSVILYVVCFIJSVIXLIWIPPIVVIGIMZIWQ SVISJJIVW

Cryptanalysis

- XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPPEVWMXMW ASVXLQSVILYVVCFIJSVIXLIWIPPIVVIGIMZIWQSVISJJIVW

- When we tabulate the frequency of letters in this ciphertext, we gets: I =14, V =13, S =12, and so on.

- The most common character is **I** with 14 occurrences. This shows that character **I** in the ciphertext probably corresponds to the character e in plaintext.

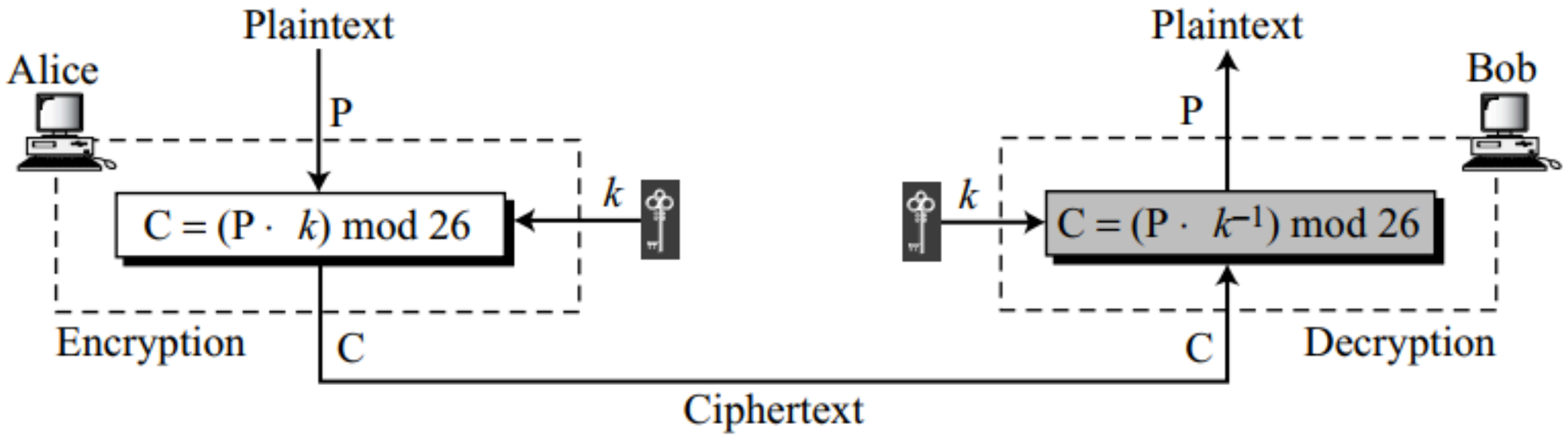
- This means key = 4.

- “the house is now for sale for four million dollars it is worth more hurry before the seller receives more offers”

Multiplicative Ciphers

- In a multiplicative cipher, the encryption algorithm specifies multiplication of the plaintext by the key and the decryption algorithm specifies division of the ciphertext by the key
- Since operations are in Z26, decryption here means multiplying by the multiplicative inverse of the key.
- The key needs to belong to the set Z26* to guarantee that the encryption and decryption are inverses of each other.

Multiplicative Ciphers



The key needs to be in Z26*.
This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.

Multiplicative Ciphers

- We use a multiplicative cipher to encrypt the message “hello” with a key of 7. The ciphertext is “XCZZU”

Plaintext: h $\rightarrow$ 07	Encryption: $(07 \times 07) \bmod 26$	ciphertext: 23 $\rightarrow$ X
Plaintext: e $\rightarrow$ 04	Encryption: $(04 \times 07) \bmod 26$	ciphertext: 02 $\rightarrow$ C
Plaintext: l $\rightarrow$ 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 $\rightarrow$ Z
Plaintext: l $\rightarrow$ 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 $\rightarrow$ Z
Plaintext: o $\rightarrow$ 14	Encryption: $(14 \times 07) \bmod 26$	ciphertext: 20 $\rightarrow$ U

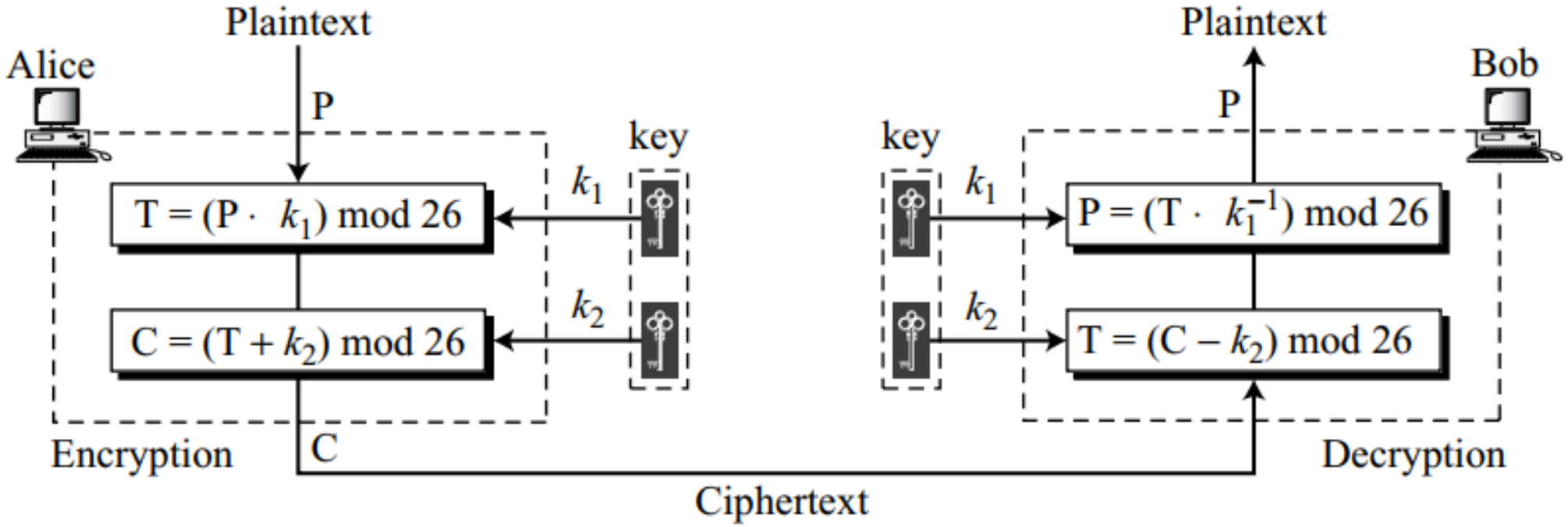
Affine Cipher

- We can combine the additive and multiplicative ciphers to get **affine cipher -** a combination of both ciphers with a **pair of keys.**

- The first key is used with the multiplicative cipher;
- the second key is used with the additive cipher

- Whenever a combination of ciphers are used, we should ensure that:
    - Each one has an inverse
    - They are used in reverse order in the encryption and decryption.
    - If addition is the last operation in encryption, then subtraction should be the first in decryption.

Affine Cipher



$$C = (P \times k_1 + k_2) \bmod 26 \qquad P = ((C - k_2) \times k_1^{-1}) \bmod 26$$

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2

Affine Cipher

- Use an affine cipher to encrypt the message “hello” with the key pair (7, 2).

Affine Cipher

- We use 7 for the multiplicative key and 2 for the additive key. We get “ZEBBW”

P: h → 07	Encryption: $(07 \times 7 + 2) \bmod 26$	C: 25 → Z
P: e → 04	Encryption: $(04 \times 7 + 2) \bmod 26$	C: 04 → E
P: l → 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 → B
P: o → 14	Encryption: $(14 \times 7 + 2) \bmod 26$	C: 22 → W

C: Z → 25	Decryption: $((25 - 2) \times 7^{-1}) \bmod 26$	P:07 → h
C: E → 04	Decryption: $((04 - 2) \times 7^{-1}) \bmod 26$	P:04 → e
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P:11 → l
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P:11 → l
C: W → 22	Decryption: $((22 - 2) \times 7^{-1}) \bmod 26$	P:14 → o

Add the additive inverse of $-2 \equiv 24 \pmod{26}$ to the received ciphertext.
Then multiply the result by the multiplicative inverse of $7^{-1} \equiv 15 \pmod{26}$ to find the plaintext characters.

Affine Cipher

- The additive cipher is a special case of an affine cipher in which k1 = 1.

- The multiplicative cipher is a special case of affine cipher in which k2 = 0.

Monoalphabetic Substitution Cipher

- Because additive, multiplicative, and affine ciphers have small key domains, they are very vulnerable to brute-force attack.

- After Alice and Bob agreed to a single key, that key is used to encrypt each letter in the plaintext or decrypt each letter in the ciphertext.

- The key is independent from the letters being transferred.

- A better solution is to create a mapping between each plaintext character and the corresponding ciphertext character.

- Alice and Bob can agree on a table showing the mapping for each character.

Monoalphabetic Substitution Cipher

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

PT : this message is easy to encrypt but hard to find the key

CT : ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

Cryptanalysis

- The size of the key space for the monoalphabetic substitution cipher is 26! (almost 4 × 1026).
- This makes a brute-force attack extremely difficult for Eve even if she is using a powerful computer.
- However, she can use statistical attack based on the frequency of characters.
- The cipher does not change the frequency of characters

Polyalphabetic Ciphers

- In polyalphabetic substitution, each occurrence of a character may have a different substitute.

- The relationship between a character in the plaintext to a character in the ciphertext is one-to-many.

- For example, “a” could be enciphered as “D” in the beginning of the text, but as “N” at the middle.

- Polyalphabetic ciphers have the advantage of hiding the letter frequency of the underlying language.

- Eve cannot use single-letter frequency statistic to break the ciphertext.

Polyalphabetic Ciphers

- To create a polyalphabetic cipher, we need to make each ciphertext character dependent on both the **corresponding plaintext character** and the **position** of the plaintext character in the message.

- This implies that **our key should be a stream of subkeys,** in which each subkey depends on the **position** of the plaintext character that uses that subkey for encipherment.

- In other words, we need to have a key stream k = (k1, k2, k3, …) in which ki is used to encipher the ith character in the plaintext to create the ith character in the ciphertext

Autokey Cipher

- To see the position dependency of the key, let us discuss a simple polyalphabetic cipher called the **autokey cipher.**

- In this cipher, the key is a stream of subkeys, in which each subkey is used to encrypt the corresponding character in the plaintext.

- The first subkey is a predetermined value secretly agreed upon by Alice and Bob.

- The second subkey is the value of the first plaintext character (between 0 and 25).

- The third subkey is the value of the second plaintext

- The name of the cipher, **autokey**, implies that the subkeys are automatically created from the plaintext cipher characters during the encryption process.

Autokey Cipher

$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3\ldots$ $k = (k_1, P_1, P_2, \ldots)$

Encryption: $C_i = (P_i + k_i) \bmod 26$ Decryption: $P_i = (C_i - k_i) \bmod 26$

Eg: If initial key value k1 = 12.

PT: “Attack is today”

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P’s Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	*12*	*00*	*19*	*19*	*00*	*02*	*10*	*08*	*18*	*19*	*14*	*03*	*00*
C’s Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	**M**	**T**	**M**	**T**	**C**	**M**	**S**	**A**	**L**	**H**	**R**	**D**	**Y**

Cryptanalysis

- The autokey cipher definitely hides the single-letter frequency statistics of the plaintext.

- However, it is still as vulnerable to the brute-force attack as the additive cipher.

- The first subkey can be only one of the 25 values (1 to 25).

- We need polyalphabetic ciphers that not only hide the characteristics of the language but also have large key domains.

Autokey Cipher

- PT : This is auto key cipher
- K1 = 20
- CT: **MOQK QK AOMC UIW EOQEII**

Playfair Cipher

- Another polyalphabetic cipher used by the British army during World War I.
- The secret key in this cipher is made of 25 alphabet letters arranged in a 5 × 5 matrix (letters I and J are considered the same when encrypting).
- Different arrangements of the letters in the matrix can create many different secret keys.
- The letters in the matrix are diagonally starting from the top right-hand corner.

Secret Key =

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

Playfair Cipher

- Before encryption, if the **two letters in a pair are the same**, a bogus letter is inserted to separate them.

- After inserting bogus letters, if the **number of characters in the plaintext is odd**, one extra bogus character is added at the end to make the number of characters even.

- The cipher uses three rules for encryption:
  - If the two letters in a pair are located in the **same row** of the secret key, the corresponding encrypted character for each letter is the **next letter to the right in the same row (**with wrapping to the beginning of the row if the plaintext letter is the last character in the row).

  - If the two letters in a pair are located in the **same column** of the secret key, the corresponding encrypted character for each letter is the **letter beneath it in the same column** (with wrapping to the beginning of the column if the plaintext letter is the last character in the column).

  - If the two letters in a pair are **not in the same row or column** of the secret, the corresponding encrypted character for each letter is a **letter that is in its own row but in the same column as the other letter**

Playfair Cipher

- The Playfair cipher meets the criteria for a polyalphabetic cipher:
    - The key is a stream of subkeys in which the subkeys are created two at a time.
    - In Playfair cipher, **the key stream and the cipher stream are the same.**

- This means that the mentioned rules can be thought of as the rules for creating the key stream.
- The encryption algorithm takes a pair of characters from the plaintext and creates a pair of subkeys by following the rules.
- We can say that the **key stream depends on the position of the character** in the plaintext.
- Position dependency: the subkey for each plaintext character depends on the next or previous neighbor.
- Looking at the Playfair cipher in this way, **the ciphertext is actually the key stream.**

Playfair Cipher

$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3 \ldots$ $k = [(k_1, k_2), (k_3, k_4), \ldots]$

Encryption: $C_i = k_i$ Decryption: $P_i = k_i$

- Ex: Encrypt the plaintext “hello” using the key given.
- When we group the letters in two-character pairs, we get “he, ll, o”.
- We need to insert an x between the two l’s (els), giving “he, lx, lo”.

Secret Key =

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

Playfair Cipher

$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3 \ldots$ $k = [(k_1, k_2), (k_3, k_4), \ldots]$

Encryption: $C_i = k_i$ Decryption: $P_i = k_i$

- Ex: Encrypt the plaintext “hello” using the key given.
- When we group the letters in two-character pairs, we get “he, ll, o”.
- We need to insert an x between the two l’s (els), giving “he, lx, lo”.

Secret Key =

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

CIPHER
- he —> EC
- lx —> QZ
- lo —> BX

Playfair Cipher

- A brute-force attack on a Playfair cipher is very difficult.
- The size of the key domain is 25! (factorial 25).
- The encipherment hides the single-letter frequency of the characters.
- But, the frequencies of diagrams are preserved (to some extent because of filler insertion), so a cryptanalyst can use a ciphertext-only attack based on the digram frequency test to find the key

Vigenere Cipher

- An interesting polyalphabetic cipher was designed by Blaise de Vigenere, a sixteenth-century French mathematician.

- A Vigenere cipher uses a different strategy to create the key stream.

- The key stream is a **repetition of an initial secret key stream of length m**, where we have **1 ≤ m ≤ 26**.

- The cipher can be described as follows where (k1, k2, …, km) is the **initial secret key** agreed to by Alice and Bob.

Vigenere Cipher

- An important difference between the Vigenere cipher and the other two polyalphabetic ciphers, is that the Vigenere key stream **does not depend on the plaintext characters**; it depends only on the **position of the character** in the plaintext.

- The key stream can be created without knowing what the plaintext is

Vigenere Cipher

$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3 \ldots$ $K = [(k_1, k_2, \ldots, k_m), (k_1, k_2, \ldots, k_m), \ldots]$

Encryption: $C_i = P_i + k_i$ Decryption: $P_i = C_i - k_i$

- Encrypt the message “She is listening” using the 6-character keyword “PASCAL”.

- The initial key stream is (15, 0, 18, 2, 0, 11)

Vigenere Cipher

Plaintext:	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P's values:	18	07	04	08	18	11	08	18	19	04	13	08	13	06
Key stream:	*15*	*00*	*18*	*02*	*00*	*11*	*15*	*00*	*18*	*02*	*00*	*11*	*15*	*00*
C's values:	07	07	22	10	18	22	23	18	11	6	13	19	02	06
Ciphertext:	H	H	W	K	S	W	X	S	L	G	N	T	C	G

Vigenere Cipher

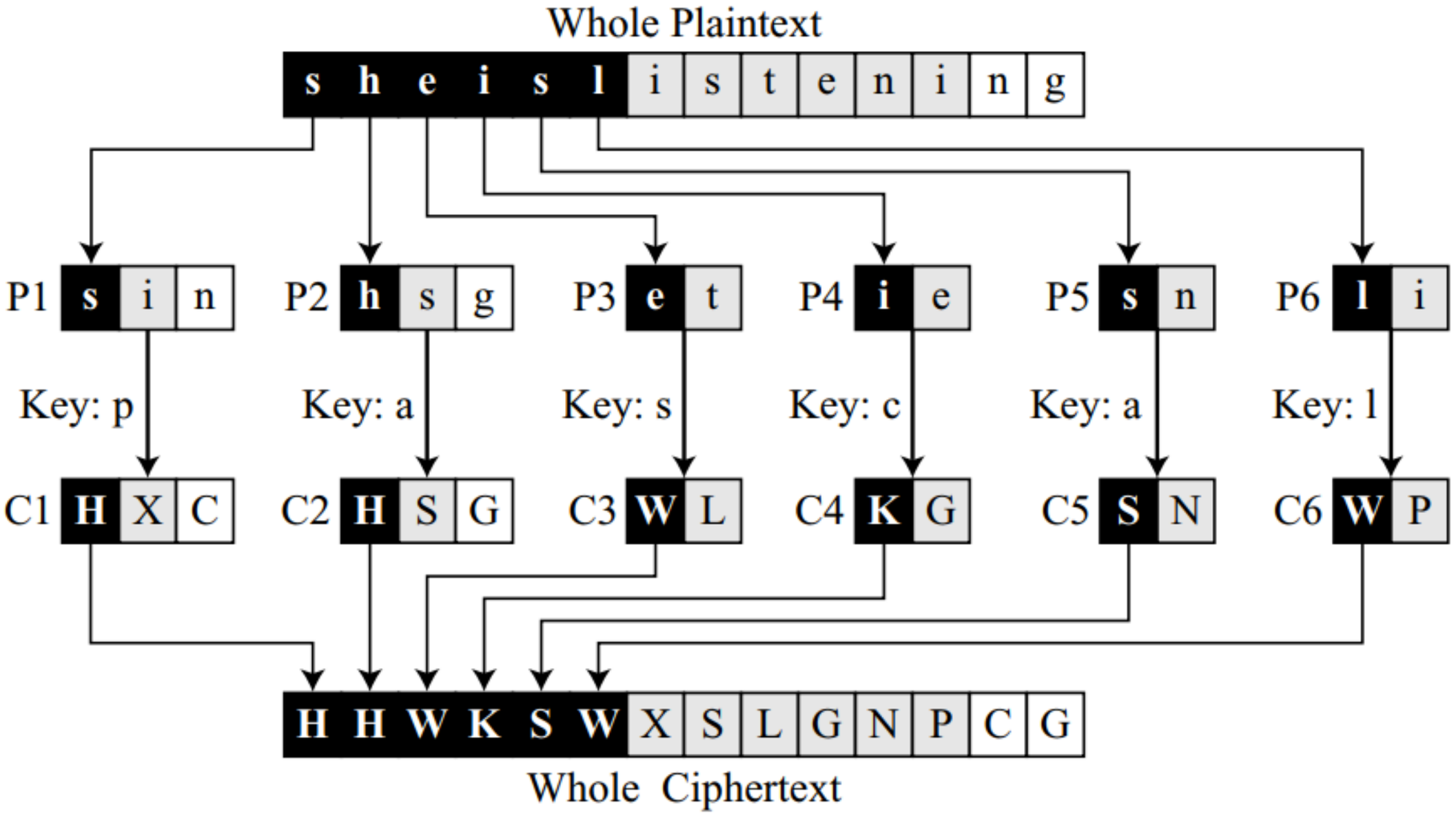
- Vigenere cipher can be seen as combinations of m additive ciphers.

- The plaintext can be thought of as six different pieces, each encrypted separately.

- There are **m pieces of the plaintext**, each encrypted with a different key, **to make m pieces of ciphertext**

- The additive cipher can be considered as a special case of Vigenere cipher in which m = 1.

Vigenere Cipher



Vigenere Tableau

- Another way to look at Vigenere ciphers is through what is called a Vigenere tableau shown in Table next slide

- The first row shows the plaintext character to be encrypted.

- The first column contains the characters to be used by the key.

- The rest of the tableau shows the ciphertext characters which can be used during encryption and decryption

Table 3.3 *A Vigenere tableau*

	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	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

Cryptanalysis of Vigenere Ciphers

- Vigenere ciphers, like all polyalphabetic ciphers, do not preserve the frequency of characters.

- However, Eve still can use some techniques to decipher an intercepted ciphertext.

- The cryptanalysis here consists of two parts:

  - finding the length of the key and

  - finding the key itself.

Cryptanalysis of Vigenere Ciphers

- **To find the length of the key :**

- Using Kasiski test, the cryptanalyst searches for **repeated text segments**, of at least **three characters**, in the ciphertext.

- Suppose that two of these segments are found and the distance between them is d.

- The cryptanalyst assumes that **d|m** where m is the key length.

- If more repeated segments can be found with distances d1, d2, …, dn, then gcd (d1, d2, …, dn)/m.

- If two characters are the same and are k × m (k = 1, 2, …) characters apart in the plaintext, they are the same and k × m characters apart in the ciphertext.

- Cryptanalyst uses segments of **at least three characters to avoid the cases where the characters in the key are not distinct.**

Cryptanalysis of Vigenere Ciphers

- We divide the ciphertext into m different pieces and applies the method used to cryptanalyze the additive cipher, including frequency attack.

- Each ciphertext piece can be decrypted and put together to create the whole plaintext.

- In other words, the whole ciphertext does not preserve the single-letter frequency of the plaintext, but each piece does.

Cryptanalysis of Vigenere Ciphers

- Cryptanalyse the following cipher text:

```
LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-
VTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFZMVVWLGYHCUSWXQH-
KVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWVWJWIXGFWLTSHGJOUEEHH-
VUCFVGOWICQLTJSUXGLW
```

Cryptanalysis of Vigenere Ciphers

- Cryptanalyse the following cipher text:

```
LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-
VTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFZMVVWLGYHCUSWXQH-
KVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWVWJWIXGFWLTSHGJOUEEHH-
VUCFVGOWICQLTJSUXGLW
```

- Kasiski test for 3 characters :

String	*First Index*	*Second Index*	*Difference*
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60
MPH	119	127	8

Cryptanalysis of Vigenere Ciphers

- The greatest common divisor of differences is 4 => the key length is multiple of 4.

- Try m = 4. Divide the ciphertext into four pieces.

- Piece C1 is made of characters 1, 5, 9, …; piece C2 is made of characters 2, 6, 10, …; and so on.

- Use the statistical attack on each piece separately.

- Interleave the decipher pieces one character at a time to get the whole plaintext.

- If the plaintext does not make sense, try with another m.

Cryptanalysis of Vigenere Ciphers

```
C1: LWGWCRAOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG
P1: jueuapymircneroarhtsthihytrahcieixsthcarrehe
C2: IGGGQHGWGKVCTSOSQSWVWFVYSHSVFSHZHWWFSOHCOQSL
P2: ussctsiswhofeaeceihcetesoecatnpntherhctecex
C3: OFDHURWQZKLZHGVVLUVLSZWHWKHFDUKDHVIWHUHFWLUW
P3: lcaerotnwhiwedssirsiirhketehretltiideatrairt
C4: MEVHCWILEMWVVXGETMEXLMLCXVELGMIMBWXLGEVVITX
P4: iardysehaisrrtcapiafpwtethecarhaesfterectpt
```

- Plain text :

Julius Caesar used a cryptosystem in his wars, which is now referred to as Caesar cipher. It is an additive cipher with the key set to three. Each character in the plaintext is shifted three characters to create ciphertext.

Hill Cipher

- Hill cipher is a polyalphabetic cipher invented by Lester S. Hill.
- Here the plaintext is divided into equal-size blocks.
- The blocks are encrypted one at a time in such a way that **each character in the block contributes to the encryption of other characters in the block**.
- The Hill cipher belongs to a category of ciphers called **block ciphers.**
- The other ciphers we studied so far belong to **stream ciphers**.

Hill Cipher

- In a Hill cipher, the key is a square matrix of size m × m in which m is the size of the block.

- If we call the key matrix K, each element of the matrix is ki,j

Hill Cipher

- In a Hill cipher, the key is a square matrix of size m × m in which m is the size of the block.

- If we call the key matrix K, each element of the matrix is $k_{i,j}$

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix}$$

Hill Cipher

- If we call the m characters in the plaintext block P1, P2, …, Pm, the corresponding characters in the ciphertext block are C1, C2, …, Cm.

$$
\begin{aligned}
C_1 &= P_1 k_{11} + P_2 k_{21} + \cdots + P_m k_{m1} \\
C_2 &= P_1 k_{12} + P_2 k_{22} + \cdots + P_m k_{m2} \\
&\cdots \\
C_m &= P_1 k_{1m} + P_2 k_{2m} + \cdots + P_m k_{mm}
\end{aligned}
$$

Hill Cipher

- The equations show that each ciphertext character such as C1 depends on all plaintext characters in the block (P1, P2, …, Pm).

- Not all square matrices have multiplicative inverses in Z26, so Alice and Bob should be careful in selecting the key.

- Bob will not be able to decrypt the ciphertext sent by Alice **if the matrix does not have a multiplicative inverse**

Hill Cipher

- The multiplicative inverse is defined only for square matrices.
- The multiplicative inverse of a square matrix A is a square matrix B such that $A \times B = B \times A = I$.
- Multiplicative inverse of A is defined by A^{-1}.
- The multiplicative inverse exists only if the det(A) has a multiplicative inverse in the corresponding set.
- Since no integer has a multiplicative inverse in Z, there is no multiplicative inverse of a matrix in Z.
- Matrices with real elements have inverses only if **det (A) ≠ 0.**

Hill Cipher

- Using matrices allows Alice to encrypt the whole plaintext.
- In this case, the plaintext is an l × m matrix in which l is the number of blocks.
- For example, the plaintext “code is ready” can make a 3 × 4 matrix when adding extra bogus character “z” to the last block and removing the spaces.
- The ciphertext is “OHKNIHGKLISS”.
- Bob can decrypt the message using the inverse of the key matrix.

Hill Cipher

$$\underset{\mathbf{C}}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} = \underset{\mathbf{P}}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} \underset{\mathbf{K}}{\begin{bmatrix} 09 & 07 & 11 & 13 \\ 04 & 07 & 05 & 06 \\ 02 & 21 & 14 & 09 \\ 03 & 23 & 21 & 08 \end{bmatrix}}$$

a. Encryption

$$\underset{\mathbf{P}}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} = \underset{\mathbf{C}}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} \underset{\mathbf{K}^{-1}}{\begin{bmatrix} 02 & 15 & 22 & 03 \\ 15 & 00 & 19 & 03 \\ 09 & 09 & 03 & 11 \\ 17 & 00 & 04 & 07 \end{bmatrix}}$$

b. Decryption

One-Time Pad

- One of the goals of cryptography is perfect secrecy.
- A study by Shannon has shown that perfect secrecy can be achieved if each plaintext symbol is encrypted with a key randomly chosen from a key domain.
- Eg: n additive cipher can be easily broken because the same key is used to encrypt every character.
- However, even this simple cipher can become a perfect cipher if the key that is used to encrypt each character is chosen randomly from the key domain (00, 01, 02, …, 25)—that is, if the first character is encrypted using the key 04, the second character is encrypted using the key 02, the third character is encrypted using the key 21; and so on.
- Ciphertext-only attack is impossible.
- Other types of attacks are also impossible if the sender changes the key each time she sends a message, using another random sequence of integers

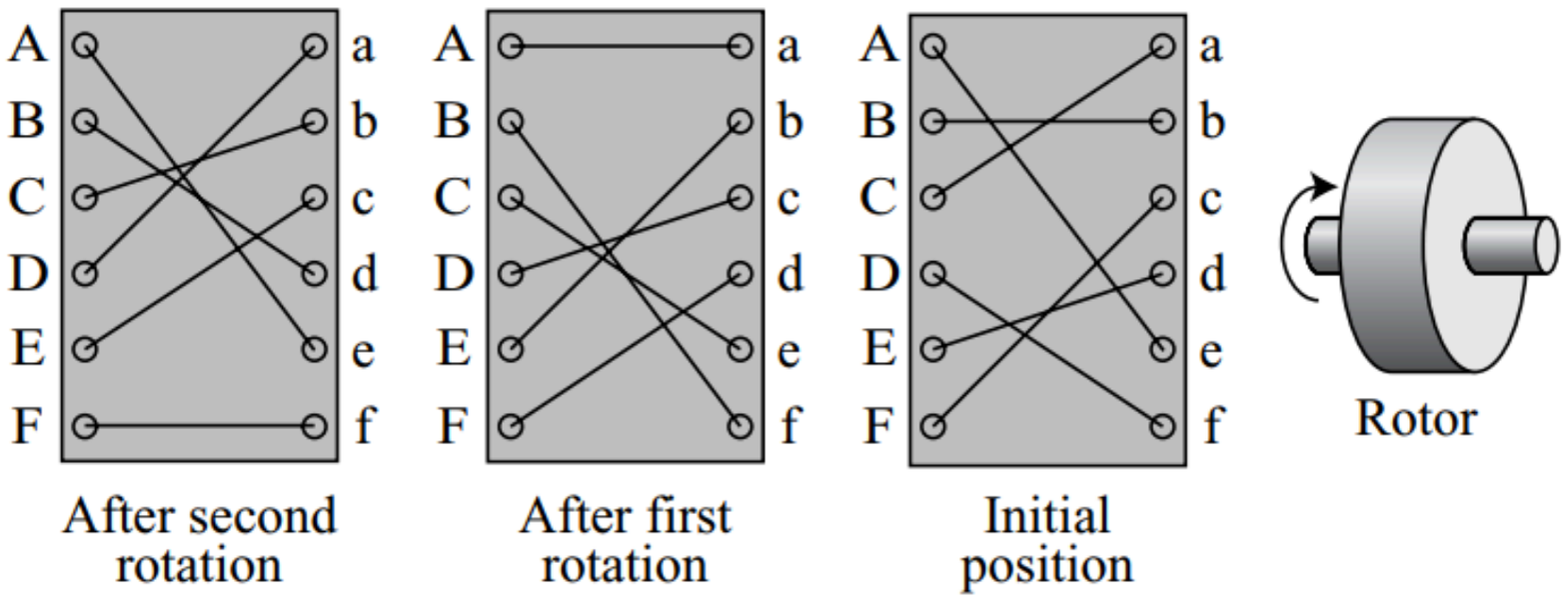
One-Time Pad

- This idea is used in a cipher called **one-time pad**, invented by Vernam.
- In this cipher, the **key has the same length as the plaintext** and is chosen completely in random.
- A **one-time pad is a perfect cipher**, but it is almost impossible to implement commercially.
- If the key must be newly generated each time, how can Alice tell Bob the new key each time she has a message to send?
- However, there are some occasions when a one-time pad can be used.
- For example, if the president of a country needs to send a completely secret message to the president of another country, she can send a trusted envoy with the random key before sending the message.

Rotor Cipher

- Although one-time pad ciphers are not practical, one step toward more secured encipherment is the **rotor cipher**.

- It uses the idea behind monoalphabetic substitution but changes the mapping between the plaintext and the ciphertext characters for each plaintext character



Rotor Cipher

- The rotor shown uses only 6 letters, but the actual rotors use 26 letters.
- The rotor is permanently wired, but the connection to encryption/ decryption characters is provided by brushes.
- The wiring is shown as though the rotor were transparent and one could see the inside.
- The initial setting (position) of the rotor is the secret key between Alice and Bob.
- The first plaintext character is encrypted using the initial setting; the second character is encrypted after the first rotation (in Figure at 1/6 turn, but the actual setting is 1/26 turn); and so on.

Rotor Cipher

- A three-letter word such as “bee” is encrypted as “BAA” if the rotor is stationary (the monoalphabetic substitution cipher), but it will encrypted as “BCA” if it is rotating (the rotor cipher).

- This shows that the rotor cipher is a polyalphabetic cipher because two occurrences of the same plaintext character are encrypted as different characters.

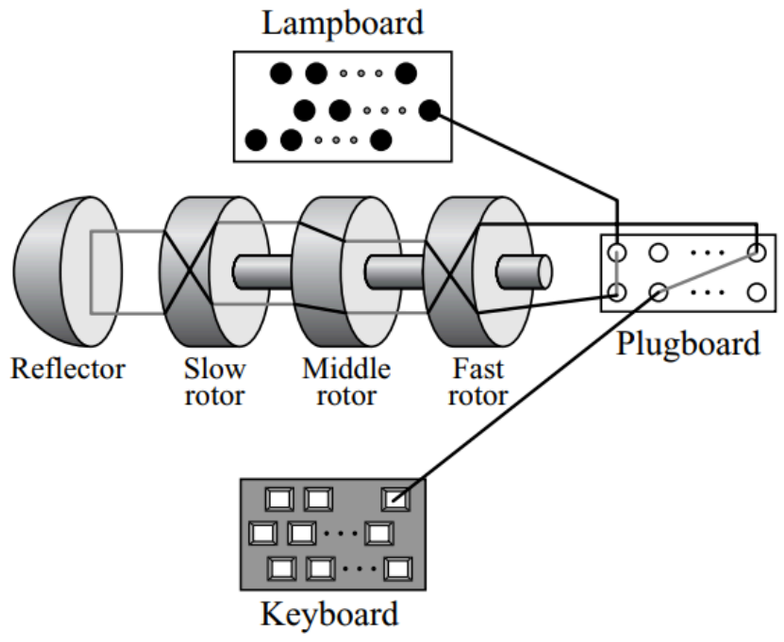
- The rotor cipher is as resistant to a brute-force attack as the monoalphabetic substitution cipher because Eve still needs to find the first set of mappings among 26! possible ones.

- The rotor cipher is much more resistant to statistical attack than the monoalphabetic substitution cipher because it does not preserve letter frequency.

Enigma Machine

- The Enigma machine was originally invented by Sherbius, but was modified by the German army and extensively used during World War II.

- The machine was based on the principle of rotor ciphers.



Enigma Machine

- The following lists the main components of the machine:
    - A keyboard with 26 keys used for entering the plaintext when encrypting and for entering the ciphertext when decrypting.
    - A lampboard with 26 lamps that shows the ciphertext characters in encrypting and the plaintext characters in decrypting.
    - A plugboard with 26 plugs manually connected by 13 wires.
    - The configuration is changed every day to provide different scrambling.
    - Three wired rotors as described in the previous section. The three rotors were chosen daily out of five available rotors.
    - The fast rotor rotates 1/26 of a turn for each character entered on the keyboard. The middle rotor makes 1/26 turn for each complete turn of the fast rotor. The slow rotor makes 1/26 turn for each complete turn of the middle rotor.
    - A reflector, which is stationary and prewired.

Enigma Machine

- **Code Book -** To use the Enigma machine, a code book was published that gives several settings for each day, including:
  - The three rotors to be chosen, out of the five available ones.
  - The order in which the rotors are to be installed.
  - The setting for the plugboard.
  - A three-letter code of the day

Enigma Machine

- **Procedure for Encrypting a Message**

- To encrypt a message, the operator followed these steps:

    - Set the starting position of the rotors to the code of the day. For example, if the code was “HUA”, the rotors were initialized to “H”, “U”, and “A”, respectively.

    - Choose a random three-letter code, such as “ACF”. Encrypt the text “ACFACF” (repeated code) using the initial setting of rotors in step 1. For example, assume the encrypted code is “OPNABT”.

    - Set the starting positions of the rotors to OPN (half of the encrypted code).

    - Append the encrypted six letters obtained from step 2 (“OPNABT”) to the beginning of the message.

    - Encrypt the message including the 6-letter code. Send the encrypted message.

Enigma Machine

- **Procedure for Decrypting a Message**
- To decrypt a message, the operator followed these steps:
    - Receive the message and separate the first six letters.
    - Set the starting position of the rotors to the code of the day.
    - Decrypt the first six letters using the initial setting in step 2.
    - Set the positions of the rotors to the first half of the decrypted code.
    - Decrypt the message (without the first six letters).

Enigma Machine

- **Cryptanalysis:**
    - The Enigma machine was broken during the war, although the German army and the rest of the world did not hear about this until a few decades later.
    - The question is how such a complicated cipher was attacked.
    - Although the German army tried to hide the internal wiring of the rotors, the Allies somehow obtained some copies of the machines.
    - The next step was to find the setting for each day and the code sent to initialize the rotors for every message.
    - The invention of the first computer helped the Allies to overcome these difficulties.
    - The full picture of the machine and its cryptanalysis can be found at some of the Enigma Websites.