BAHRIA UNIVERSITY
Discovering Knowledge

Substitution Techniques

Data Encryption & Security (CEN-451)

Spring 2025 (BSE-8A&B)

Substitution Technique

- Techniques in which the letters of plaintext are replaced by other letters, numbers or symbols.
- If the plaintext is viewed as a **sequence of bits**, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.
- Substitution ciphers can be categorized as either **monoalphabetic ciphers** or **polyalphabetic ciphers**.

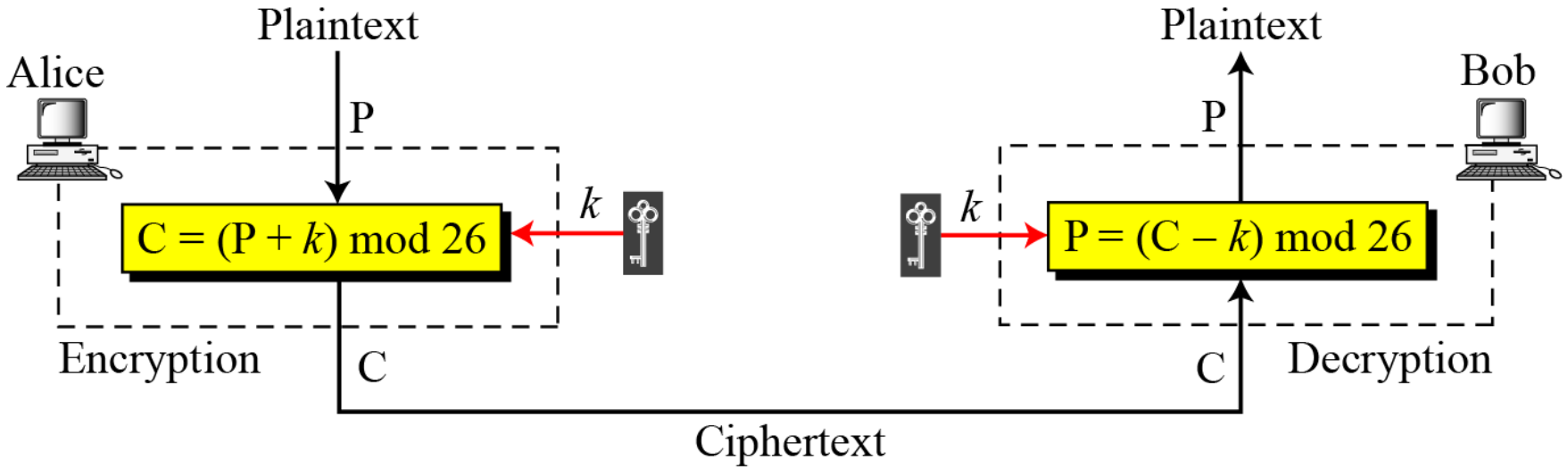
Monoalphabetic Ciphers

- In **monoalphabetic** substitution, the relationship between a symbol in the plaintext to a symbol in the ciphertext is always **one-to-one**.
- **Example:** the following shows a plaintext and its corresponding ciphertext. The cipher is **monoalphabetic** because both ***l's*** are encrypted as ***O's***.

Plaintext: hello **Ciphertext:** KHOOR

Additive 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25



Additive Cipher (Cont.)

- The plaintext, ciphertext and key are integers in Z_{26}.
- The general algorithm is:

$$C = E(k, P) = (P + k) \bmod 26$$

- The decryption algorithm is simply:

$$P = D(k, C) = (C - k) \bmod 26$$

- Where, k takes on a value in the range 1 to 25).
- With $k = 0$, ciphertext is same as plaintext. Hence, only 25 keys are useful.

Additive Cipher (Cont.)

- **Example 01:** use the additive cipher with $k = 15$ to encrypt the message "hello".
- **Solution:**

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

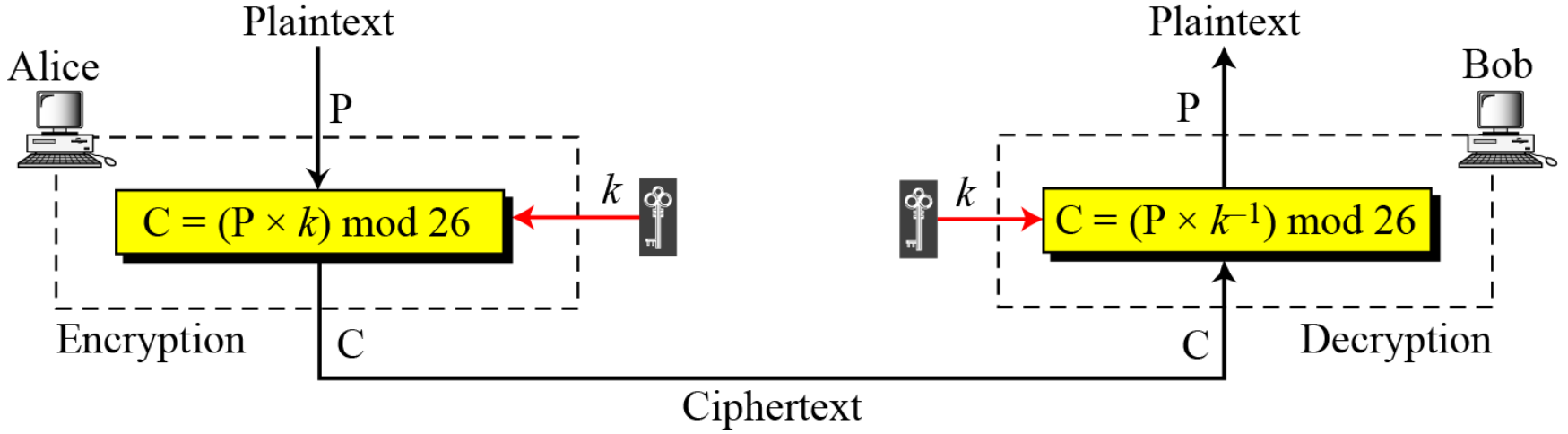
Additive Cipher (Cont.)

- **Example 02:** use the additive cipher with $k = 15$ to decrypt the message "WTAAD".
- **Solution:**

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

Multiplicative Ciphers

- **Multiplicative cipher** is a **monoalphabetic**.
- The plaintext and ciphertext are integers in Z_{26}, the key is an integer in Z_{26}^*.
- Encryption is multiplying plain text by key, while decryption is multiplying ciphertext by multiplication inverse of that key.



Multiplicative Ciphers (Cont.)

- **Example:** Encrypt the message "hello" using the multiplicative cipher with a key of 7.

- **Solution:**

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

- The ciphertext is **"XCZZU"**.

Multiplicative Ciphers (Cont.)

- **Example:** Decrypt the ciphertext "XCZZU" into its original form, provided that encryption key is 7 in $\mathbf{Z_{26}}$.

Find the multiplicative inverse of 7 in Z_{26}.

$\mathbf{r = r_1 - (q \times r_2)}$ $\mathbf{t = t_1 - (q \times t_2)}$

q	r1	r2	r	t1	t2	t
	26	7		0	1	

Brute-Force in Additive Cipher

- If it is known that a given ciphertext is an Additive cipher, then brute-force is easily performed.
- ***Simply try all the 25 possible keys.***
- In this example, the plaintext leaps out as occupying the third line.
- ***Note:*** with only 25 possible keys, the Additive cipher is far from secure.

KEY	PHHW PH DIWHU WKH WRJD SDUWB
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufs uif uphb qbsuz
3	meet me after the toga party
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxoqv
7	iaap ia wbpan pda pkcw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby niau julns
10	fxxm fx tymxk max mhzt itkmr
11	ewwl ew sxlwj lzw lgys hsjlq
12	dvvk dv rwkvi kyv kfxr grikp
13	cuuj cu qvjuh jxu jewq fqhjo
14	btti bt puitg iwt idvp epgin
15	assh as othsf hvs hcuo dofhm
16	zrrg zr nsgre gur gbtn cnegl
17	yqqf yq mrfqd ftq fasm bmdfk
18	xppe xp lqepc esp ezrl alcej
19	wood wo kpdob dro dyqk zkbdi
20	vnnc vn jocna cqn cxpj yjach
21	ummb um inbmz bpm bwoi xizbg
22	tlla tl hmaly aol avnh whyaf
23	skkz sk glzkx znk zumg vgxze
24	rjjy rj fkyjw ymj ytlf ufwyd
25	qiix qi ejxiv xli xske tevxc

Polyalphabetic Ciphers

- In polyalphabetic substitution, each occurrence of a character may have a different substitute.
- The relationship between a character in plaintext to a character in ciphertext is **one-to-many**.
- **E.g.** "a" could be enciphered as "D" at beginning of text, but as "N" at the middle.
- **Benefit of polyalphabetic ciphers:**
  - Hides letter frequency of the language.
  - Cannot use frequency statistics to break the ciphertext.

Playfair Cipher

- The Playfair algorithm is based on **5 × 5** matrix of letters.
- The **secret key** is made of **25 alphabet letters** arranged in **5 × 5** matrix, where letters **I/J** are considered the same when encrypting.
- Different arrangements of letters in the matrix can create many different **secret keys**.
- **Example of a secret key:**

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 (Cont.)

- In Playfair key matrix; fill in letters of a given **keyword *(minus duplicates)*** from left to right and top to bottom, then fill rest of the matrix with the remaining letters in alphabetic order.
- **Example:** using the keyword **MONARCHY**

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

Playfair Cipher (Cont.)

- Playfair treats **digrams** in the plaintext as **"single units"** and translates these units into ciphertext **digrams**, according to the following rules:
  1. If both letters are the same, a bogus letter is inserted to separate them.
  2. After inserting bogus letters, if number of characters is **odd**, then add one extra character at the end. Hence, number of characters is **even**.

Playfair Cipher (Cont.)

- Playfair cipher rules (Cont.):
  3. If both letters fall in the same row, then replace each with letter to the right (wrapping back to start from end). **E.g. "ar"** encrypts as **"RM"**.
  4. If both letters fall in the same column, then replace each with the letter below it (wrapping to top from bottom). **E.g. "mu"** encrypts to **"CM"**.
  5. Otherwise, each letter is replaced by the one in its row in the column of the other letter of the pair. **E.g. "hs"** encrypts to **"BP"**, and **"ea"** to **"IM"** or **"JM"** ***(as desired)***.

Playfair Cipher (Cont.)

- **Example:** encrypt the plaintext "hello" using the key

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

- **Solution:**

he → EC lx → QZ lo → BX

Plaintext: hello Ciphertext: ECQZBX

Playfair Cipher: Cryptanalysis

- Brute-force attack on Playfair is difficult, since size of key domain is **25!**
- Encipherment hides the single-letter frequency of the character.
- Though there are only 26 letters in the secret key, but there are **26 * 26 = 676** digrams. Hence, identification of individual digrams is more difficult.
- However, frequency of diagrams are preserved. Hence, cryptanalysis can use diagram frequency text to find the key.

Hill Cipher

- The plainttext is divided into **equal-size blocks** that are encrypted one at a time, where m is the size of the block.
- Each character in a block contributes to the encryption of other characters in the block.
- Hill cipher algorithm takes m successive plaintext letters and **substitutes** by m ciphertext letters.
- The key is a square matrix of size $m \times m$.

Hill Cipher (Cont.)

- Key in Hill cipher:

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

- The key matrix needs to have a **multiplicative inverse**, where not all square matrices do in $\mathbf{Z_{26}}$.

Hill Cipher (Cont.)

- Substitution is determined by ***m*** linear equations.
- If we call ***m*** characters in plaintext block $P_1, P_2, \ldots, P_m$, the corresponding characters in ciphertext block are $C_1, C_2, \ldots, C_m$.
- ***Each ciphertext character depends on all plaintext characters***.

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

Hill Cipher (Cont.)

- **E.g.** for $m = 3$, the system can be described as:

$$c_1 = (k_{11}p_1 + k_{21}p_2 + k_{31}p_3) \bmod 26$$

$$c_2 = (k_{12}p_1 + k_{22}p_2 + k_{32}p_3) \bmod 26$$

$$c_3 = (k_{13}p_1 + k_{23}p_2 + k_{33}p_3) \bmod 26$$

- This can be expressed in terms of row vectors and matrices as:

$$(c_1\ c_2\ c_3) = (p_1\ p_2\ p_3)\begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \bmod 26$$

$$\mathbf{C} = \mathbf{PK} \bmod 26$$

Hill Cipher (Cont.)

- **Example 01:** consider the plaintext **"paymoremoney"** and use the encryption key for encryption purpose through Hill cipher, where $m = 3$.

$$\mathbf{K} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

Solution:

- The first three letters of the plaintext are represented by the vector (15 0 24).
- Then (15 0 24)**K** = (303 303 531) mod 26 = (17 17 11) = RRL.

Hill Cipher (Cont.)

Solution (Cont.):

- Continuing this way, the ciphertext for the entire plaintext is RRLMWBKASPDH.
- Decryption requires using the inverse of the matrix **K**.

$$\mathbf{K}^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

- It can be seen that if the matrix $\mathbf{K}^{-1}$ is applied to the ciphertext, then the plaintext is recovered.

Hill Cipher (Cont.)

- **Example 02:** the plaintext "code is ready" can make a 3 × 4 matrix, where ***m*** = 4, when adding extra bogus character "z" to the last block along with removing spaces.
- Ciphertext is:

  "OHKNIHGKLISS"
- ***Note:*** (mod 26) is applied.

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

a. Encryption

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

b. Decryption

Hill Cipher (Cont.)

Hill Cipher Benefits:

- It completely hides single-letter frequencies.
- The use of a larger matrix hides more frequency information.
- A Hill cipher hides not only single-letter but also two-letter frequency information.
- Hill cipher is strong against a **ciphertext-only** attack.

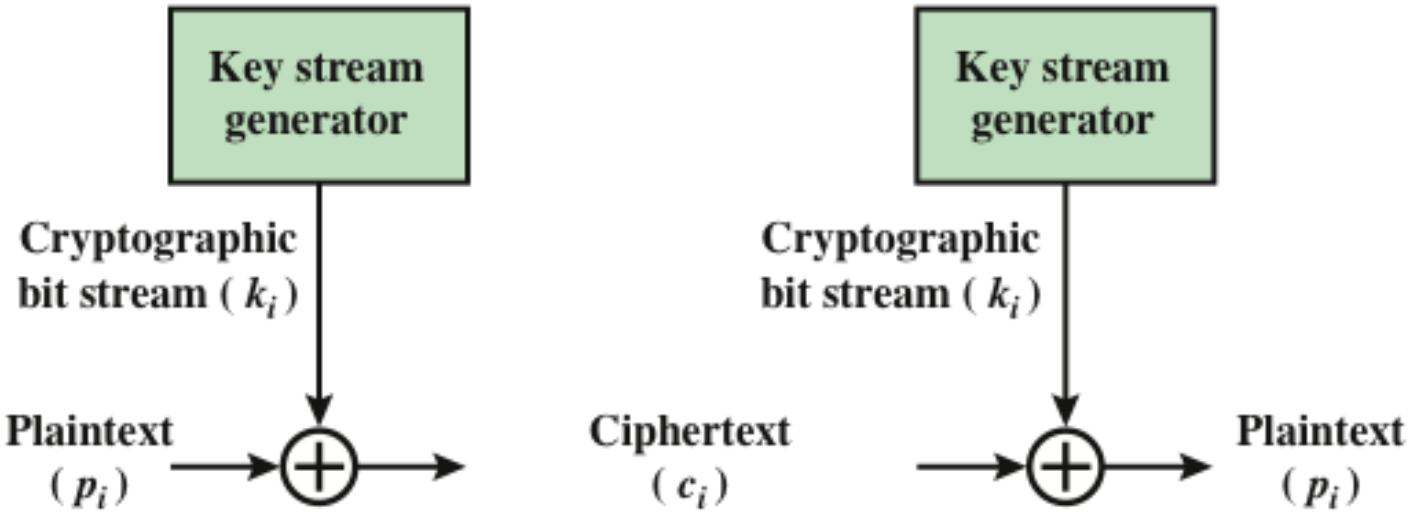
Hill Cipher Cryptanalysis

- Hill cipher is easily broken with a **known plaintext** attack.
- We define two $m \times m$ matrices $\mathbf{X} = (p_{ij})$ and $\mathbf{Y} = (c_{ij})$. Then, we can form the matrix equation $\mathbf{Y} = \mathbf{XK}$.
- If $\mathbf{X}$ has an inverse, then we can determine $\mathbf{K} = \mathbf{X}^{-1}\mathbf{Y}$.
- If $\mathbf{X}$ is not invertible, then a new version of $\mathbf{X}$ can be formed with additional plaintext–ciphertext pairs until an invertible $\mathbf{X}$ is obtained.

Vernam Cipher

- Choose a keyword that is as long as the plaintext and has no statistical relationship to it.
- The system works on **binary data (bits)** rather than letters. The system can be expressed as follows:

p_i = ith binary digit of plaintext
k_i = ith binary digit of key
c_i = ith binary digit of ciphertext
$\oplus$ = exclusive-or (XOR) operation



Vernam Cipher (Cont.)

- The ciphertext is generated by performing the bitwise XOR of the plaintext and the key.
- Because of the properties of the XOR, decryption simply involves the same bitwise operation:

$$p_i = c_i \oplus k_i$$

- Essence of this technique is the means of key construction.

One-Time Pad

- An improvement to the Vernam cipher.
- 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.

One-Time Pad (Cont.)

- **Advantage of One-Time Pad:**
  - Produces random output that bears no statistical relationship to the plaintext.
  - The ciphertext contains no information whatsoever about the plaintext, hence there is simply no way to break the code.
- **Disadvantage of One-Time Pad:**
  - Problem of making large quantities of random keys.
  - Key distribution problem, where for every message, a key of equal length is needed at both sender and receiver.

Thank You!