Traditional/Classical Encryption Techniques

G.P. Biswas
Prof/CSE, IIT-ISM, Dhanbad

Symmetric Encryption

A symmetric ciphering can be represented mathematically as

$$\textit{Encryption:} \quad C = \mathrm{E}_K(P)$$
$$\textit{Decryption:} \quad P = \mathrm{D}_K(C)$$

where P = plaintext, C = ciphertext, K = secret key, E = encryption algorithm, D = decryption algorithm and both E and D are public

Traditional Ciphers

- Substitution Ciphers : A substitution cipher replaces one symbol with another. Some of the well-known classical ciphers are-
  - Caesar cipher
  - Affine cipher
  - Monoalphabetic /Polyalphabetic Ciphers
  - One time pad
- Transposition Ciphers: A transposition cipher reorders/rearranges (permutes) symbols

Caesar/Additive Cipher

- Invented by Julius Caesar in world war II for text document and the earliest known substitution cipher.
- Each letter is replaced by the letter three positions ahead of English alphabet.
  - Encryption/decryption are

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: 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

- Example: ISM Dhanbad → LVP GKDQEDG

Contd.

- For mathematical representation, English letters mapped to numbers as

```
a, b, c, ..., x,  y,  z
0, 1, 2, ..., 23, 24, 25
```

- The Caesar cipher is:

$$C = E_K(P) = (P + k) \bmod 26$$

$$P = D_K(C) = (C - k) \bmod 26$$

Cryptanalysis

- Objective: To recover plaintext and/or secret key (here, key is fixed and $k = 3$) of a ciphertext.
- Kerckhoff's principle: the adversary knows all details about a cryptosystem except the secret key.

Two general approaches:

- brute-force attack
- non-brute-force attack (cryptanalytic attack)

Cryptanalysis of Caesar Cipher

- Not enough keys.
  - If we shift a letter 26 times, we get the same letter back and a shift of *27* is the same as a shift of *1*.
  - So we only have 25 keys (1 to 25) and

    Key space: {0, 1, …, 25}
- An attacker Eve just tries every key until she finds the right one so vulnerable to brute-force attacks, which is trivial here.

Multiplicative Cipher

- The plaintext is encrypted by multiplication operation with mod *26*
- The decryption algorithm specifies division of the ciphertext by the key or by multiplying with multiplicative inverse of the key
- Thus, decryption is not always possible for a key, whose multiplicative inverse with mod 26 does not exist.

Multiplicative cipher

- The encryption is

$$C = (K \times P) \bmod 26$$

- The Decryption is

$$P = (K^{-1} \times C) \bmod 26$$

- Here, decryption is possible if K is coprime with modulo i.e. *gcd*(*K,26*)=*1, ex. gcd(2, 26)= 2* and *gcd(3, 26) = 1. So, 2* cannot be a key, however, *3* can be and multiplicative inverse of 3 with mod 26 is 9 as 3×9 ≡ 1 mod 26.

Modular Arithmetic

We know Z denotes the set of integers from negative infinity to positive infinity as

$$\mathbf{Z} = \{ \ldots, -2, -1, 0, 1, 2, \ldots \}$$

The modulo operation creates a set, which in modular arithmetic is referred to as the set of **least residues modulo n, or Z_n**.

$$\mathbf{Z}_n = \{ 0, 1, 2, 3, \ldots, (n-1) \}$$

$$\mathbf{Z}_2 = \{ 0, 1 \}$$

$$\mathbf{Z}_6 = \{ 0, 1, 2, 3, 4, 5 \}$$

$$\mathbf{Z}_{11} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$$

In Z_n, two numbers a and b are the multiplicative inverse of each other if

$$a \times b \equiv 1 \pmod{n}$$

Multiplicative Inverse

- The key domain for any multiplicative cipher under modulo 26:

1,3,5,7,9,11,15,17,19,21,23,25

Number	1	3	5	7	9	11	15	17	19	21	23	25
Multiplicative inverse	1	9	21	15	3	19	7	23	11	5	17	25

Affine Cipher

- It uses both additive shift cipher and multiplicative cipher combinedly
- The Affine cipher is defined as follows:
  - Encryption and decryption are:

$$C = (\alpha P + \beta) \bmod 26$$

$$\gcd(\alpha, 26) = 1$$

$$P = (\alpha^{-1}(C - \beta)) \bmod 26$$

Cryptanalysis of Affine Cipher

- The key for this encryption method is the pair (α, β).
- There are 12 possible choice for α with $\gcd(\alpha,26)=1$
- 26 choices for β .
- Total choice is $12 \times 26 = 312$ and the security level of affine cipher is *312*, which is individually more than shift (which is 26) and multiplicative ciphers (12).

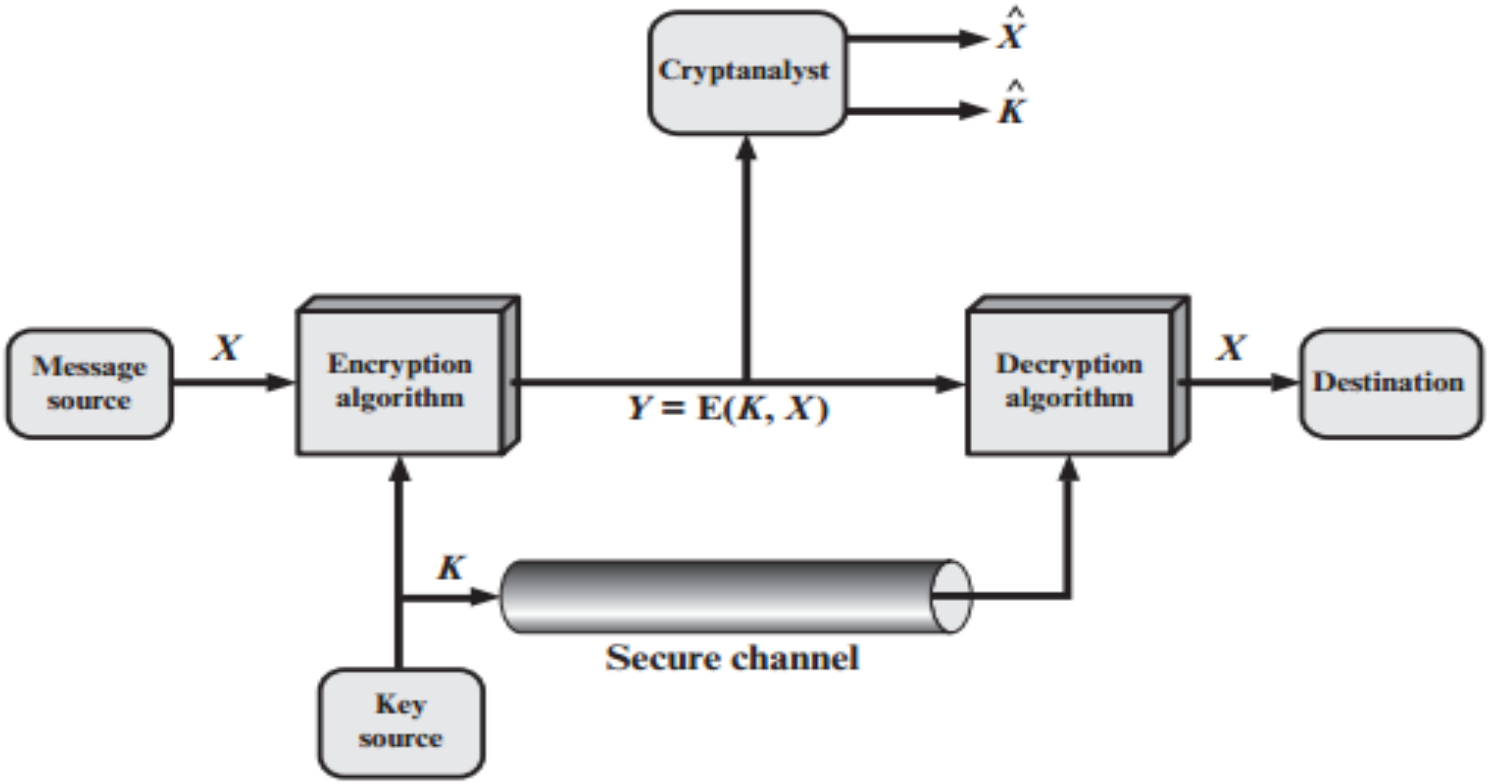


Figure 3.2 Model of Symmetric Cryptosystem

Table 3.1 Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext Only	■ Encryption algorithm ■ Ciphertext
Known Plaintext	■ Encryption algorithm ■ Ciphertext ■ One or more plaintext–ciphertext pairs formed with the secret key
Chosen Plaintext	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	■ Encryption algorithm ■ Ciphertext ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

- Known plaintext attack:
  - Consider plaintext is *if* and ciphertext is *PG* , *i.e.,*

    $P(15) = 8(i) \times k1 + k2 \bmod 26$ and

    $G(6) = 5(f) \times k1 + k2 \bmod 26$

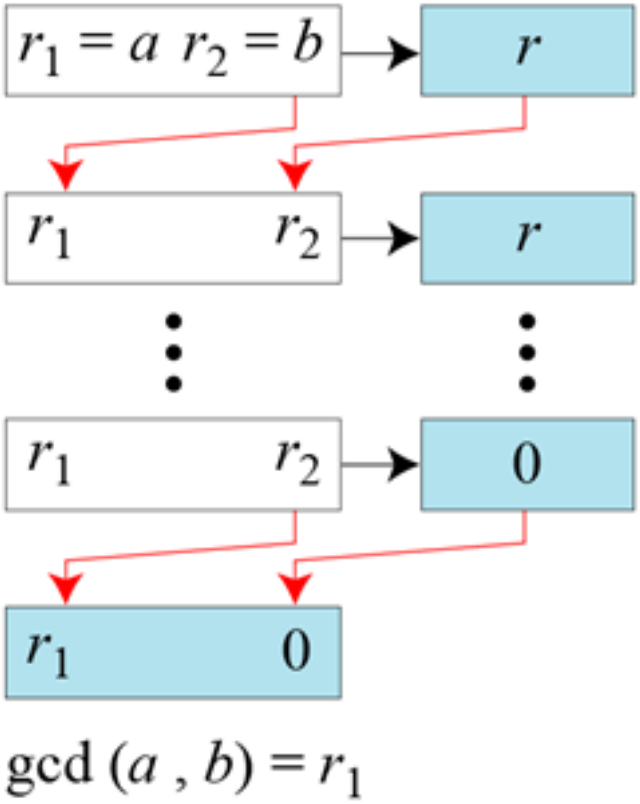
  - If $k1 = 3$ and $k2 = 17$, the above two congruents are satisfied (and they can be calculated easily).

Euclidean Algorithm

Rule 1: gcd (a, 0) = a

Rule 2: gcd (a, b) = gcd (b, r)

where r is the remainder of dividing a by b



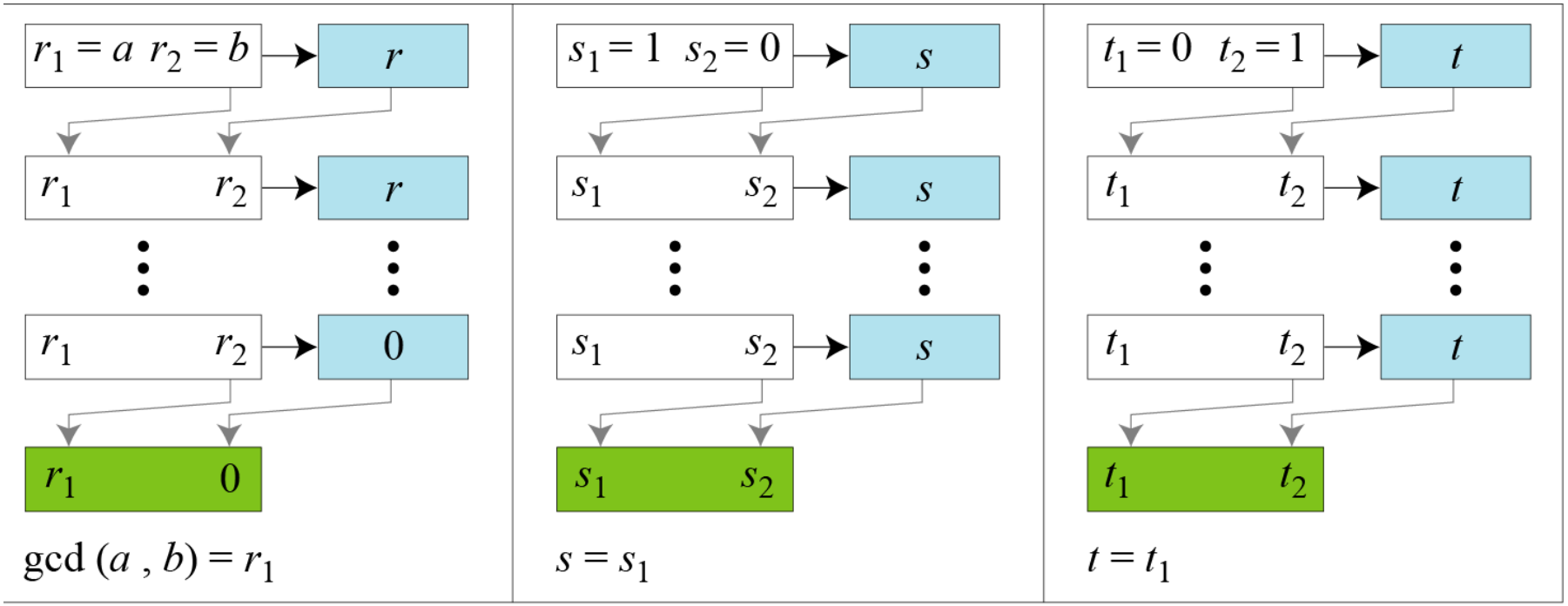
Extended Euclidean Algorithm

Given two integers a and b, we often need to find other two integers, s and t, such that

$$s \times a + t \times b = \gcd(a, b)$$

The extended Euclidean algorithm can calculate the gcd(a,b) and at the same time calculate the value of s and t.

Contd…



where

$$r \leftarrow r_1 - q \times r_2 ; \quad \text{(Updating } r\text{'s)}$$

$$s \leftarrow s_1 - q \times s_2 ; \quad \text{(Updating } s\text{'s)}$$

$$t \leftarrow t_1 - q \times t_2 ; \quad \text{(Updating } t\text{'s)}$$

Example

Given a = 161 and b = 28, find gcd (a, b) and the values of s and t.

We get gcd (161, 28) = 7, s = −1 and t = 6.

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	−5
1	28	21	7	0	1	−1	1	−5	6
3	21	7	0	1	−1	4	−5	6	−23
	7	0		**−1**	4		**6**	−23	

Multiplicative inverse

$$s \times a + t \times b = \gcd(a, b)$$

$(s \times n) + (t \times a) = \gcd(n,a) = 1$

$((s \times n) + (t \times a)) \bmod n = 1 \bmod n$

$(0 + (t \times a)) \bmod n = 1$

$(t \times a) \bmod n = 1$

This means t is the multiplicative inverse of a

Congruence

- Two numbers ***a*** and ***b*** are said to be "*congruent modulo **n***" if

  a mod n = b mod n

  implies ***a ≡ b(mod n)***

- The difference between ***a*** and ***b*** will be a multiple of ***n***

  So ***a-b = kn*** for some value of ***k***

If $a \equiv 0 \pmod{n}$, then $n \mid a$.

Properties of Congruences

1. $a \equiv b \pmod{n}$ *if* $n|(a-b)$
2. $a \equiv b \pmod{n}$ *implies* $b \equiv a \pmod{n}$
3. $a \equiv b \pmod{n}$ *and* $b \equiv c \pmod{n}$ **imply** $a \equiv c \pmod{n}$

Proof of 1.

If $n|(a-b)$, then $(a-b) = kn$ for some k. Thus, we can write $a = b + kn$. Therefore,

$(a \bmod n)$ = (remainder when $b + kn$ is divided by n) = (remainder when b is divided by n) = $(b \bmod n)$.

Modular Arithmetic

Claim: If $a \equiv b \bmod m$ and $c \equiv d \bmod m$

$$ac \equiv bd \bmod m$$

Proof: $a - b = k_1 m$ and $c - d = k_2 m$

$$ac = (b + k_1 m)(d + k_2 m) =$$

$$= bd + (k_1 d + k_2 b)m + k_1 k_2 m^2 =$$

$$= bd + (k_1 d + k_2 b + k_1 k_2 m)m$$

$$ac - bd = (k_1 d + k_2 b + k_1 k_2 m)m$$

Monoalphabetic Substitution Cipher

- Rather than having a fixed shift change, every plaintext letter to be an arbitrary ciphertext letter and prepare a table as shown.
- This table is securely exchanged with receiver.
- To decrypt we just look up the ciphertext letter in the table and then write down the matching plaintext letter

Plaintext	Ciphertext
a	G
b	X
c	N
d	S
e	D
…	…
z	Q

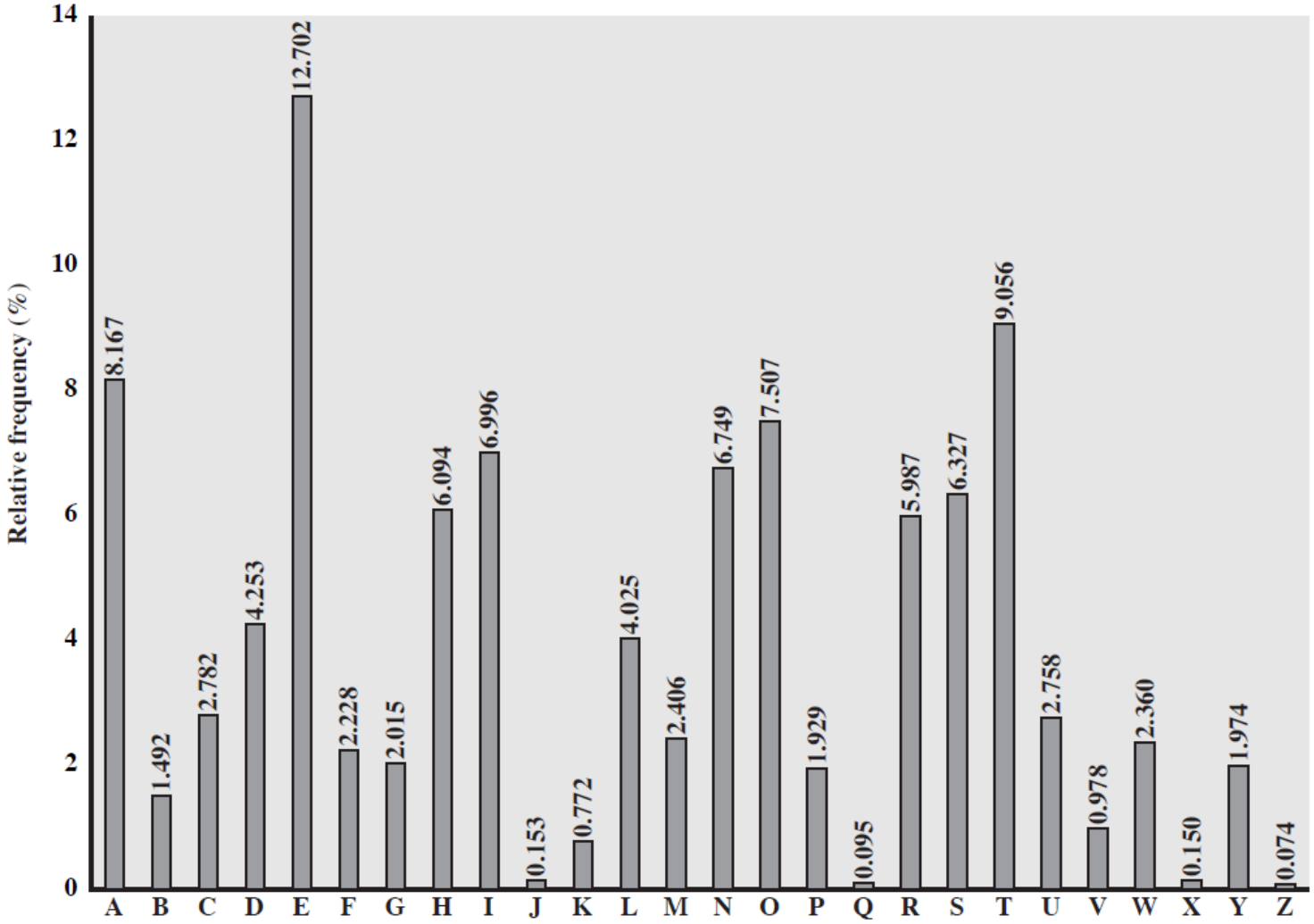
Security complexity

- Now we have a total of 26! ≈ 4 x 10^{26} keys, which is much more than in Cesar, Additive, Multiplicative and Affine ciphers.
- It is secure against brute-force attacks.
- But it is not secure against Frequency Analysis attack

Frequency Analysis

- Human languages are not random.
- In English (or any language) certain letters are used more often than others.
- In English, E is by far the most common letter, followed by T, R, N, I, O, A, S.
- If we look at a ciphertext, certain ciphertext letters are going to appear more often than others.
- It would be a good guess that the letters that occur most often in the ciphertext are actually the most common English letters as mentioned here.

General trend for English Letter Frequencies



Statistics for double & triple letters

- In decreasing order of frequency

Double letters:

th he an in er re es on, …

Triple letters:

the and ent ion tio for nde, …

Frequency Analysis Attack

- Count relative letter frequencies in a ciphertext.
- Compare this distribution against the known one.
- For example, there is a good chance that the most frequently occurred character from ciphertext will map corresponds to e, and so on.
- Proceeding with trial and error to finally get the probable plain text.

Frequency Analysis in Practice

- Consider the following ciphertext:

  dq lqwurgxfwlrq wr frpsxwlqj surylglqj d eurdg vxuyhb ri wkh glvflsolqh dqg dq lqwurgxfwlrq wr surjudpplqj. vxuyhb wrslfv zloo eh fkrvhq iurp: ruljlqv ri frpsxwhuv, gdwd uhsuhvhqwdwlrq dqg vwrudjh, errohdq dojheud, gljlwdo orjlf jdwhv, frpsxwhu dufklwhfwxuh, dvvhpeohuv dqg frpslohuv, rshudwlqj vbvwhpv, qhwzrunv dqg wkh lqwhuqhw, wkhrulhv ri frpsxwdwlrq, dqg duwlilfldo lqwhooljhqfh.

- This is an assignment to you to apply Frequency Analysis attack to above ciphertext and get the plaintext.

Playfair Cipher

- Not even the large number of keys in a monoalphabetic cipher provides security.
- One approach to improving security is to encrypt multiple letters at a time.
- The **Playfair Cipher** is the best known such cipher.

Playfair Key Matrix

- Use a *5 x 5* matrix.
- Fill in with letters of a key (without duplicates).
- Fill the rest of matrix with other remaining letters.
- Ex., let key = *playfair*

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

Encrypting and Decrypting

Plaintext is encrypted two letters at a time.

1. If a pair is a repeated letter, insert filler like 'X’.
2. If both letters fall in the same row, replace each with the letter to its right (circularly).
3. If both letters fall in the same column, replace each with the letter below it (circularly).
4. Otherwise, each letter is replaced by the letter in the same row but in the column of the other letter of the pair.

Example

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

et becomes MN

me becomes EG

qa becomes WB

Security of Playfair Cipher

- It security level is 25! (lesser than Monoalphabetic, 26!)
- It **can** be broken as it still leaves some structure of plaintext intact.
- Thus, unless the keyword is long, the last few rows of the matrix are predictable.

Polyalphabetic Ciphers

- The mono alphabetic ciphers do not change the frequency of characters in the cipher text.
- It improves security using multiple cipher alphabets for a plaintext letter.
- It makes cryptanalysis harder with more alphabets/plaintext letter and avoids frequency analysis attack.
- Use a key to select which alphabet is used for each letter of the message

Autokey Cipher

- The key is a stream of subkeys, in which each subkey is used to encrypt the corresponding plaintext letter.
- The first subkey is a predetermined value secretly agreed upon by sender and receiver.
- The second subkey is the value of the first plaintext letter, third subkey is the value of second plaintext letter and so on until the given plaintext is encrypted.

Example of Autokey Cipher

- Let plaintext is attack and the pre-considered secret key is *12*
- The ciphertext are-

  0 (a)+12 = (12)M

  19(t)+0 (a) = (19) T

  19(t)+19 = (12) M

  0(a)+19 = (19) T

  2(c) +0 = (2) C

  10(k) + 2 =(12) M

- Ciphertext is MTMTCM

Vigenère Cipher

- It is simplest polyalphabetic substitution cipher
- Consider a plaintext P
- Take a secret key: ex. security
- Encrypt each letter of *P* using security in turn.
- Repeat key until entire plaintext *P* is encrypted.
- Decryption simply works in reverse direction

Example of Vigenère Cipher

- Keyword: *deceptive*

```
key:           deceptivedeceptivedeceptive
plaintext: wearediscoveredsaveyourself
ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ
```

Security of Vigenère Ciphers

- There are multiple (how many?) ciphertext letters corresponding to each plaintext letter.
- So, letter frequencies are obscured but not totally lost.
- To break Vigenere cipher:

1. Try to guess the key length. How?
2. If key length is *N*, the cipher consists of *N* Caesar ciphers. Plaintext letters at positions *k, N+k, 2N+k, 3N+k,* etc., are encoded by the same key letter.
3. Attack each individual cipher as before.

Hill Cipher

- It is another example of a polyalphabetic cipher.
- The sender and receiver must first agree upon a key matrix *A* of size *n* x *n*.
- The key must be invertible under mod 26.
- Encryption:

$$C = \left(K_{n\times n} \times P\right) \bmod 26$$

- Decryption:

$$P = \left(K_{n\times n}{}^{-1} \times C\right) \bmod 26$$

3.1 Encryption with the Hill Cipher

The Hill Cipher Encryption Algorithm

1. Find an $n \times n$ matrix E that is invertible modulo 26. This is actually the encryption key.

2. Take the message that is to be sent (the plaintext), remove all of the spaces and punctuation symbols, and convert the letters into all uppercase.

3. Convert each character to a number between 0 and 25. The usual way to do this is $A = 0, B = 1, C = 2, \ldots, Z = 25$.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

As a historical note, Lester Hill did not use this coding of letters to numbers, he simply mixed up the order. Mixing up the order does not make the method more secure, it simply combines the Hill cipher with a simple substitution cipher, which are easy to break.

4. Divide this string of numbers up into blocks of size n. Note that if E is an $n \times n$ matrix then the block size is n. Another note, if the message does not break evenly into blocks of size n we pad the ending of the message with characters, this can be done at random.

5. Write each block as a column vector of size n. At this point the message is a sequence of n-dimensional vectors, $v_1, v_2, \ldots, v_t$.

6. Take each of the vectors and multiply them by the encryption matrix E, so

$$\begin{aligned} Ev_1 &= w_1 \\ Ev_2 &= w_2 \\ Ev_3 &= w_3 \\ &\vdots \\ Ev_t &= w_t \end{aligned}$$

7. Take the vectors $w_1, w_2, \ldots, w_t$, write the entries of the vectors in order, convert the numbers back to characters and you have your ciphertext.

One note about this algorithm is that we can do step 6 with a single matrix multiplication. If we let the message matrix M be the matrix produced by having the vectors $v_1, v_2, \ldots, v_t$ as columns, that is, $M = [v_1 \ v_2 \ \ldots \ v_t]$ then $EM = [w_1 \ w_2 \ \ldots \ w_t] = C$ would be our ciphertext matrix.

Example 7: Say Alice wants to send Bob the message "Cryptography is cool!"

1. Alice chooses the block size $n = 3$ and chooses the encryption matrix E to be,

$$E = \begin{bmatrix} 2 & 3 & 15 \\ 5 & 8 & 12 \\ 1 & 13 & 4 \end{bmatrix}$$

   Since $\det(E) \pmod{26} = 11$, and 11 is invertible modulo 26, the matrix E is also invertible modulo 26.

2. The message that is to be sent is "Cryptography is cool!", removing the spaces and punctuation symbols, and convert the letters into all uppercase gives

   CRYPTOGRAPHYISCOOL

3. Conversion to numbers using $A = 0$, $B = 1$, $C = 2$, $\ldots$, $Z = 25$, gives

   2 17 24 15 19 14 6 17 0 15 7 24 8 18 2 14 14 11

4. Dividing this string of numbers up into blocks of size 3.

   2 17 24 15 19 14 6 17 0 15 7 24 8 18 2 14 14 11

   so no padding is needed here.

5. Converting these blocks into a message matrix M gives,

$$M = \begin{bmatrix} 2 & 15 & 6 & 15 & 8 & 14 \\ 17 & 19 & 17 & 7 & 18 & 14 \\ 24 & 14 & 0 & 24 & 2 & 11 \end{bmatrix}$$

6. Multiply by the encryption matrix E,

$$EM = \begin{bmatrix} 2 & 3 & 15 \\ 5 & 8 & 12 \\ 1 & 13 & 4 \end{bmatrix} \begin{bmatrix} 2 & 15 & 6 & 15 & 8 & 14 \\ 17 & 19 & 17 & 7 & 18 & 14 \\ 24 & 14 & 0 & 24 & 2 & 11 \end{bmatrix} = \begin{bmatrix} 25 & 11 & 11 & 21 & 22 & 1 \\ 18 & 5 & 10 & 3 & 0 & 2 \\ 7 & 6 & 19 & 20 & 16 & 6 \end{bmatrix} = C$$

7. Convert C into the ciphertext.

25 18 7 11 5 6 11 10 19 21 3 20 22 0 16 1 2 6

ZSHLFGLKTVDUWAQBCG

So Alice will send "ZSHLFGLKTVDUWAQBCG" to Bob.

The Hill Cipher Decryption Algorithm

1. Find $D = E^{-1} \pmod{26}$. This is the decryption key.
2. Take the ciphertext and convert it to the matrix C.
3. Calculate $DC = M$.
4. Convert the matrix M to the plaintext message. You may need to insert the appropriate spaces and punctuation symbols since these were removed.

Example 8: Bob has the encrypted message ZSHLFGLKTVDUWAQBCG.

1. He calculates

$$\begin{bmatrix} 2 & 3 & 15 \\ 5 & 8 & 12 \\ 1 & 13 & 4 \end{bmatrix}^{-1} \pmod{26} = \begin{bmatrix} 10 & 19 & 16 \\ 4 & 23 & 7 \\ 17 & 5 & 19 \end{bmatrix}$$

2. He also converts the ciphertext to the matrix C.

ZSHLFGLKTVDUWAQBCG

25 18 7 11 5 6 11 10 19 21 3 20 22 0 16 1 2 6

and since he knows that the block size is 3 he constructs C as

$$C = \begin{bmatrix} 25 & 11 & 11 & 21 & 22 & 1 \\ 18 & 5 & 10 & 3 & 0 & 2 \\ 7 & 6 & 19 & 20 & 16 & 6 \end{bmatrix}$$

3. Calculate $DC = M$.

$$DC = \begin{bmatrix} 10 & 19 & 16 \\ 4 & 23 & 7 \\ 17 & 5 & 19 \end{bmatrix} \begin{bmatrix} 25 & 11 & 11 & 21 & 22 & 1 \\ 18 & 5 & 10 & 3 & 0 & 2 \\ 7 & 6 & 19 & 20 & 16 & 6 \end{bmatrix} = \begin{bmatrix} 2 & 15 & 6 & 15 & 8 & 14 \\ 17 & 19 & 17 & 7 & 18 & 14 \\ 24 & 14 & 0 & 24 & 2 & 11 \end{bmatrix} = M$$

4. Convert the matrix M to the plaintext message.

2 17 24 15 19 14 6 17 0 15 7 24 8 18 2 14 14 11

CRYPTOGRAPHYISCOOL

So Bob adds in a couple spaces to get CRYPTOGRAPHY IS COOL!

Transposition Ciphers

- Also called **permutation** ciphers.
- Shuffle the plaintext, without altering the actual letters used.
- Example: Row Transposition Ciphers

Row Transposition Ciphers

- Plaintext is written row by row in a rectangle.
- Ciphertext: write out the columns in an order specified by a key.

Key: 3 4 2 1 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: TTNAAPTMTSUOAODWCOIXKNLYPETZ

One Time Pad (OTP) Cipher

- An Army Signal Corp officer, Joseph Mauborgne, proposed an improvement to the Vernam cipher that yields the ultimate in security.
  - Mauborgne suggested using a random key that is as long as the message, so that the key need not be repeated.
  - In addition, the key is to be 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. Such a scheme, known as a one-time pad, is unbreakable.
  - It 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.

An example should illustrate our point. Suppose that we are using a Vigenère scheme with 27 characters in which the twenty-seventh character is the space character, but with a one-time key that is as long as the message. Consider the ciphertext

```
ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
```

We now show two different decryptions using two different keys:

```
ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
key:        pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih
plaintext:  mr mustard with the candlestick in the hall

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
key:        pftgpmiydgaxgoufhkllmhsqdqogtewbqfgyovuhwt
plaintext:  miss scarlet with the knife in the library
```

Suppose that a cryptanalyst had managed to find these two keys. Two plausible plaintexts are produced. How is the cryptanalyst to decide which is the correct decryption (i.e., which is the correct key)? If the actual key were produced in a truly random fashion, then the cryptanalyst cannot say that one of these two keys is more likely than the other. Thus, there is no way to decide which key is correct and therefore which plaintext is correct.

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	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

One Time Pad (OTP)

- It is a classical (additive) cipher
  - Key is chosen randomly
  - Plaintext $X = (x_1\ x_2\ \dots\ x_n)$
  - Key $K = (k_1\ k_2\ \dots\ k_n)$
  - Ciphertext $Y = (y_1\ y_2\ \dots\ y_n)$

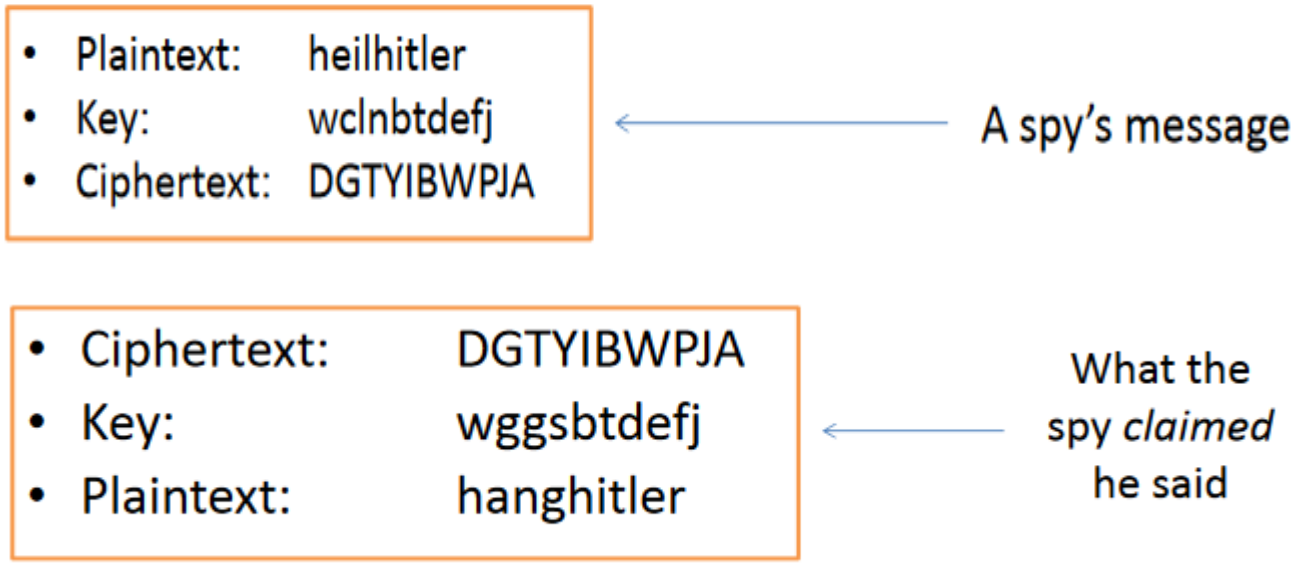
  - $e_k(X) = (x_1+k_1\ \ x_2+k_2\ \dots\ \ x_n+k_n) \bmod m$
  - $d_k(Y) = (y_1-k_1\ \ y_2-k_2\ \dots\ \ y_n-k_n) \bmod m$
- Intuitively, it is perfectly secure
- Key is random, the ciphertext too will be completely random

OTP Contd.

- Basic Idea: Ciphertext should provide no "information" about Plaintext (Shannon theory, 1949). We say such a scheme has perfect secrecy and thus, One-time pad has perfect secrecy.
- One-Time Pad uses a very long key, what if the key is not chosen randomly, instead, texts from, e.g., a book is used.
  - This is not One-Time Pad anymore–this does not have perfect secrecy–this can be broken
- The key in One-Time Pad should never be reused.–If it is reused, it is Two-Time Pad, and is insecure.

OTP Example

- One-Time Pad was used in World War II: one-time key material was printed on silk, which agents could conceal inside their clothing; whenever a key had been used, it was torn off and burnt.



- Key is changed and both plaintexts are meaningful, and guessing of correct one is difficult, and that is why OTP is perfectly secure

In fact, given any plaintext of equal length to the ciphertext, there is a key that produces that plaintext. Therefore, if you did an exhaustive search of all possible keys, you would end up with many legible plaintexts, with no way of knowing which was the intended plaintext. Therefore, the code is unbreakable.

The security of the one-time pad is entirely due to the randomness of the key. If the stream of characters that constitute the key is truly random, then the stream of characters that constitute the ciphertext will be truly random. Thus, there are no patterns or regularities that a cryptanalyst can use to attack the ciphertext.

In theory, we need look no further for a cipher. The one-time pad offers complete security but, in practice, has two fundamental difficulties:

1. 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. Supplying truly random characters in this volume is a significant task.
2. Even more daunting is the problem of key distribution and protection. For every message to be sent, a key of equal length is needed by both sender and receiver. Thus, a mammoth key distribution problem exists.

Because of these difficulties, the one-time pad is of limited utility and is useful primarily for low-bandwidth channels requiring very high security.

The one-time pad is the only cryptosystem that exhibits what is referred to as *perfect secrecy*. This concept is explored in Appendix F.