UNIT 2

Classical Cryptography

Content…

- Introduction
- Encryption Method

  1. Symmetric/private(secret) key/one key/same key
  2. Asymmetric/public key/ key pair/different key

- Cryptography

  1.Operation Used:- Substitution & Transposition

  2.*Key*:-

  3.*Type of processing*:- Algorithm types are block & stream

  4. *No. of Rounds*

- Substitution
- Transposition

Product Cipher

Terminology

- Code
  - Replacement based on words or semantic structures
- Cipher
  - Replacement based on symbol
- Cryptosystem
  - A model used for encryption & dencryption process

Terminology

- Cryptography
  - The science of encrypting or hiding secrets.
- Cryptanalysis
  - The art & science of decrypting messages or breaking codes and ciphers.
- Cryptanalyst
  - The hacker/attacker/person who attempts to break.
- Cryptology
  - The combination of the two.

Terminology

- Plaintext – an unencrypted message
- Cipher text – an encrypted message
- Key – pattern of alphabets & numbers
- Security: - a combination of
  - Authentication
  - Authorization
  - Access control

Three Era's of Cryptology

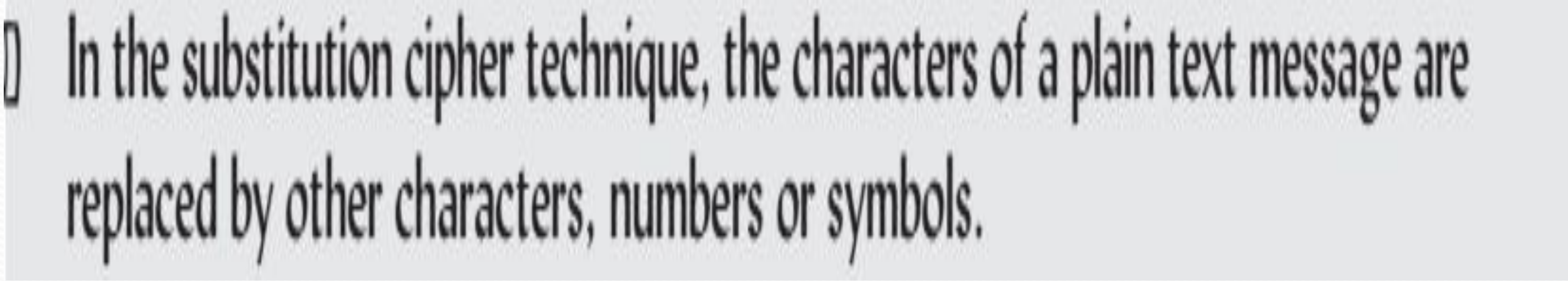
- Pre-World War II
  - Cryptography as a craft
  - Widely used, but few provable techniques
- 1940s-1970
  - Secret key encryption introduced
  - Information theory used to characterize security
- 1970-present
  - Public key systems introduced

Substitution Ciphers

- Plain text are replaced by other character, number, symbols according to a key.

  - Substitution is said to add *confusion*
    - Measure of the relationship between plaintext and cipher text

SUBSTITUTION TECHNIQUES



- In the substitution cipher technique, the characters of a plain text message are replaced by other characters, numbers or symbols.

- Caesar Cipher
- Modified Version of Caesar Cipher
- Mono-alphabetic Cipher
- Homophonic Substitution Cipher
- Polygram Substitution Cipher
- Polyalphabetic Substitution Cipher
- Playfair Square
- Hill

Early cryptography

- Caesar cipher
  - Julius Caesar in 100 B.C
  - Replace each letter l with $(l+3) \bmod 26$
  - “Attack at dawn” becomes
  - Dwwdfn dw gdzq

Plain text

Cipher Text

- Two components:
  - Algorithm: Shift characters by a fixed amount
  - Key: the fixed amount.
  - *Uniform Scheme* of substitution

Caesar Cipher

- The Caesar cipher is still useful as a way to prevent people from *unintentionally reading something*.

- Fundamental problem: key length is shorter than the message.

- Weak Scheme:- work backwards/reverse to break this.

- Note: Knowing the algorithm (but not the key) makes this cipher much easier to crack

  - 26 possibilities v/s 26!

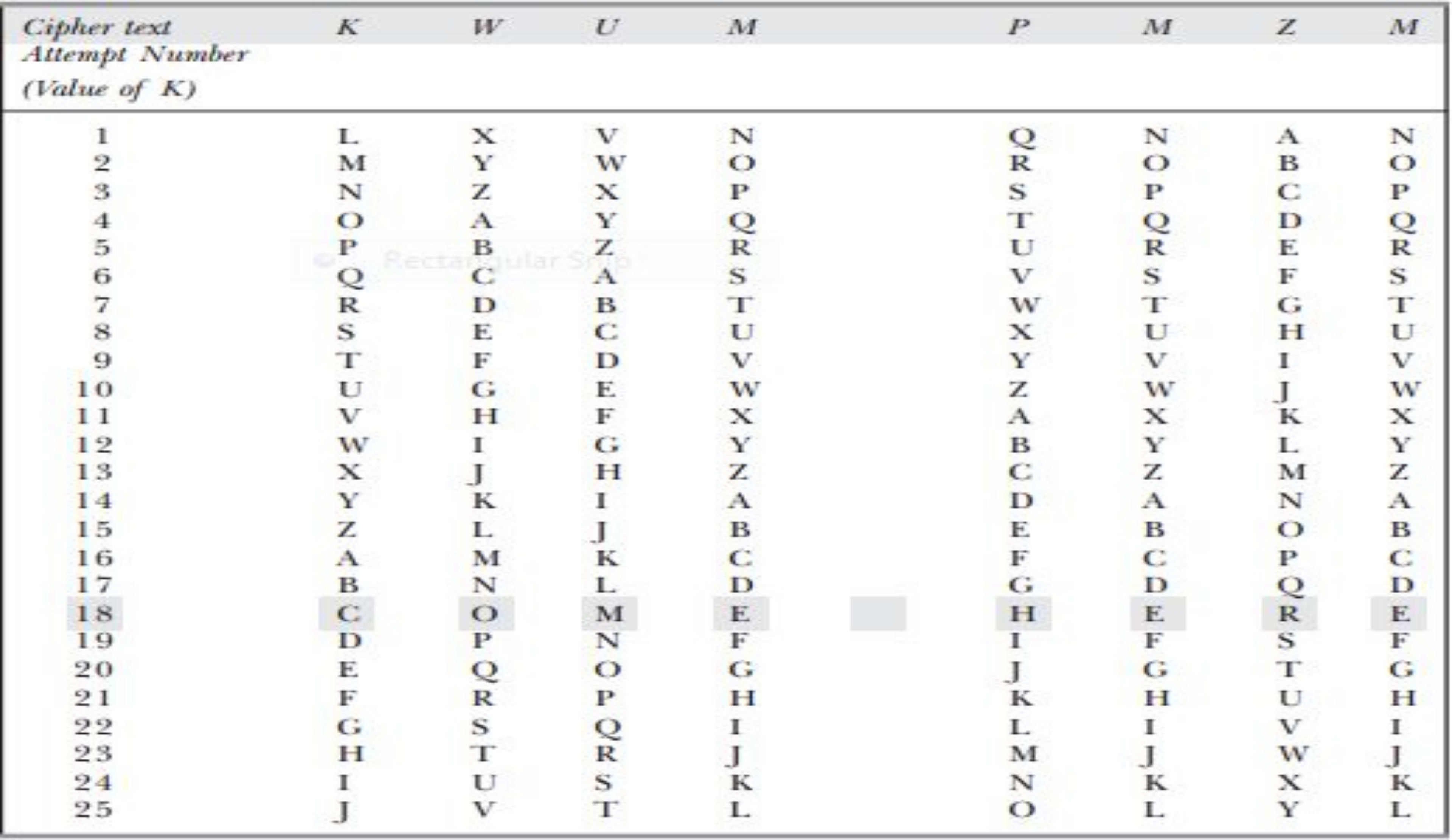
Modified Version of Caesar Cipher

- It cab be k *place* down the line this can increase complexity.
- Once replacement scheme is decided, it would be constant for whole PT.
- 25 possibilities of replacement.
- Brute force attack
- Measure Weakness *Predictability*
- Refer table on next slide

Weaknesses of the Caesar Cipher

- Word structure is preserved.
  - Break message into equal-length blocks.
    - dww dfn dwg dzq
- Solution: use multiple keys
  - E.g. shift by (3,5,7)
    - “Attack at dawn” becomes dya dhr dyk dbu
    - Better, but frequency information still present.
    - An attacker that knows the block size can separate out characters encoded with different keys.

Attempt to Break Modified Caesar Cipher Text Using All Possibilities



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

Mono-alphabetic Cipher

- *RondomScheme* of substitution

Change

- No relation between the replacement of one to other.
- High number of possible permutation & combination.
- Mathematically (26*25*24*23*…………………..2) or 4*10^26
- If the cipher text created with this technique is *short*
- Letter frequency is a big clue
  - e,t,a,o most common English letters.
  - Using a *single key* preserves frequency.

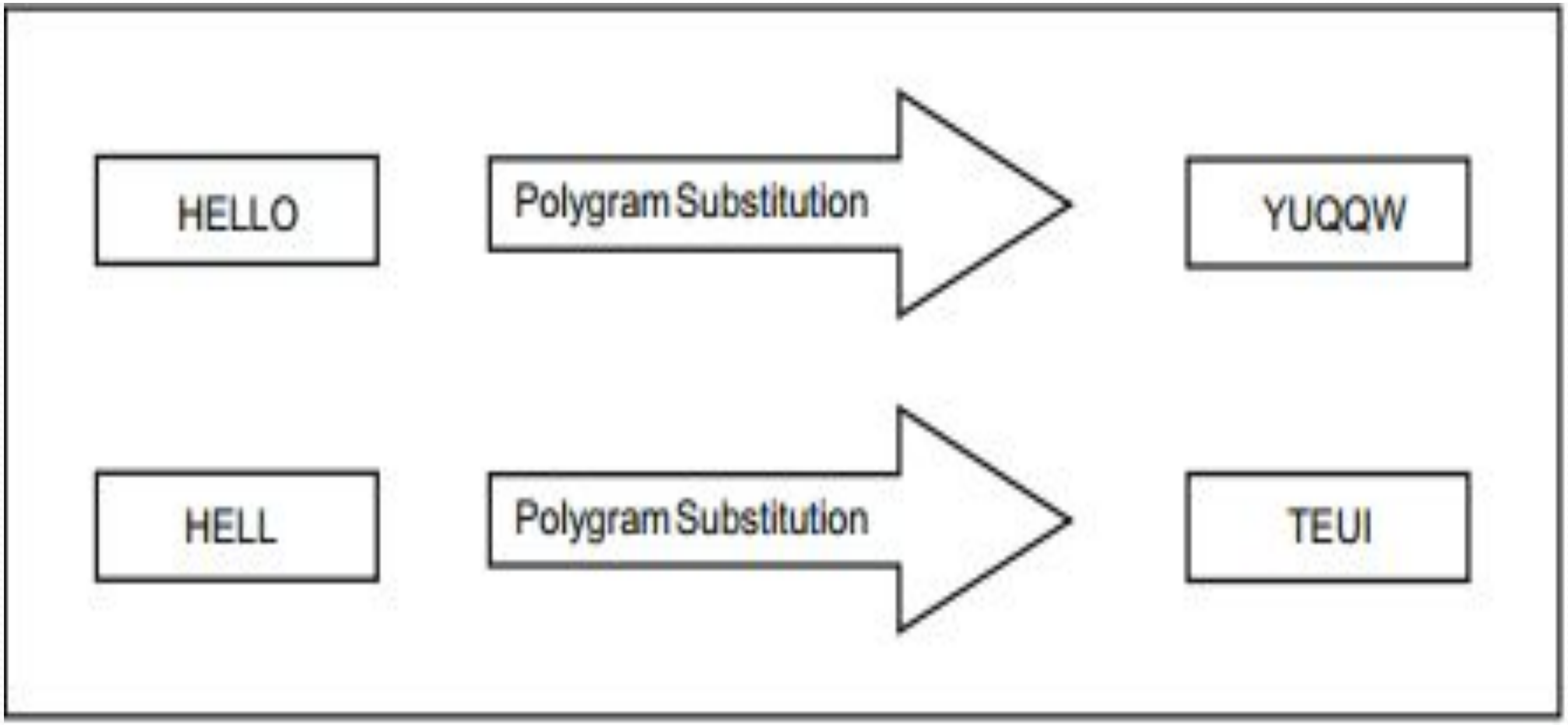
- There is only one hitch. If the cipher text created with this technique is short, the cryptanalyst can try different attacks based on her knowledge of the English language.
- As we know, some alphabets in the English language occur more frequently than others.
- Language analysts have found that given a single alphabet in cipher text, the probability
- that it is a P is 13.33%—the highest.
- After P comes Z, which is likely to occur 11.67%.
- The probability that the alphabet is C, K, L, N or R is almost 0—the lowest.
- A cryptanalyst looks for patterns of alphabets in a cipher text, substitutes the various
- available alphabets in place of cipher text alphabets, and then tries her attacks.
- Apart from single-alphabet replacements, the cryptanalyst also looks for repeated patterns of words to try the attacks. For example, the cryptanalyst might look for two-alphabet cipher text patterns since the word to occurs very frequently in English.
- If the cryptanalyst finds that two alphabet combinations are found frequently in a cipher text message, she might try and replace all of them with to, and then try and deduce the remaining alphabets/words.
- Next, the cryptanalyst might try to find repeating three-alphabet patterns and try and replace them with the word the, and so on.

Homophonic substitution cipher

- To escape frequency analysis, we can use a homophonic substitution cipher
  - Map symbols to multiple symbols.
  - *e.g* 0 -> {01, 10}, 1->{00,11}
  - Given Text 011010010 becomes: 011100101101011110
  - Advantage: -Frequencies Hidden
  - Disadvantage: -Message and Key are Longer

Polygram Cipher

- All previous techniques are based on *stream cipher.*
- Polygram is a *block cipher.*
- *Following figure shows it*



Poly-alphabetic Cipher

- *Leon Battista* invented in 1568.
- This cipher has been broken many times, and yet it has been used extensively.
- The Vigenere Cipher and the Beaufort Cipher are examples of it.
- It uses *multiple one-character keys*.
- Each of the keys encrypts one plain text character.
- The first key encrypts the first plain text character; the second key encrypts the second plain text character, and so on.
- After all the keys are used, they are recycled.
- Thus, if we have 26 one-letter keys, every 26th character in the plain text would be replaced with the same key.
- This number (in this case, 26) is called as the *period* of the cipher.

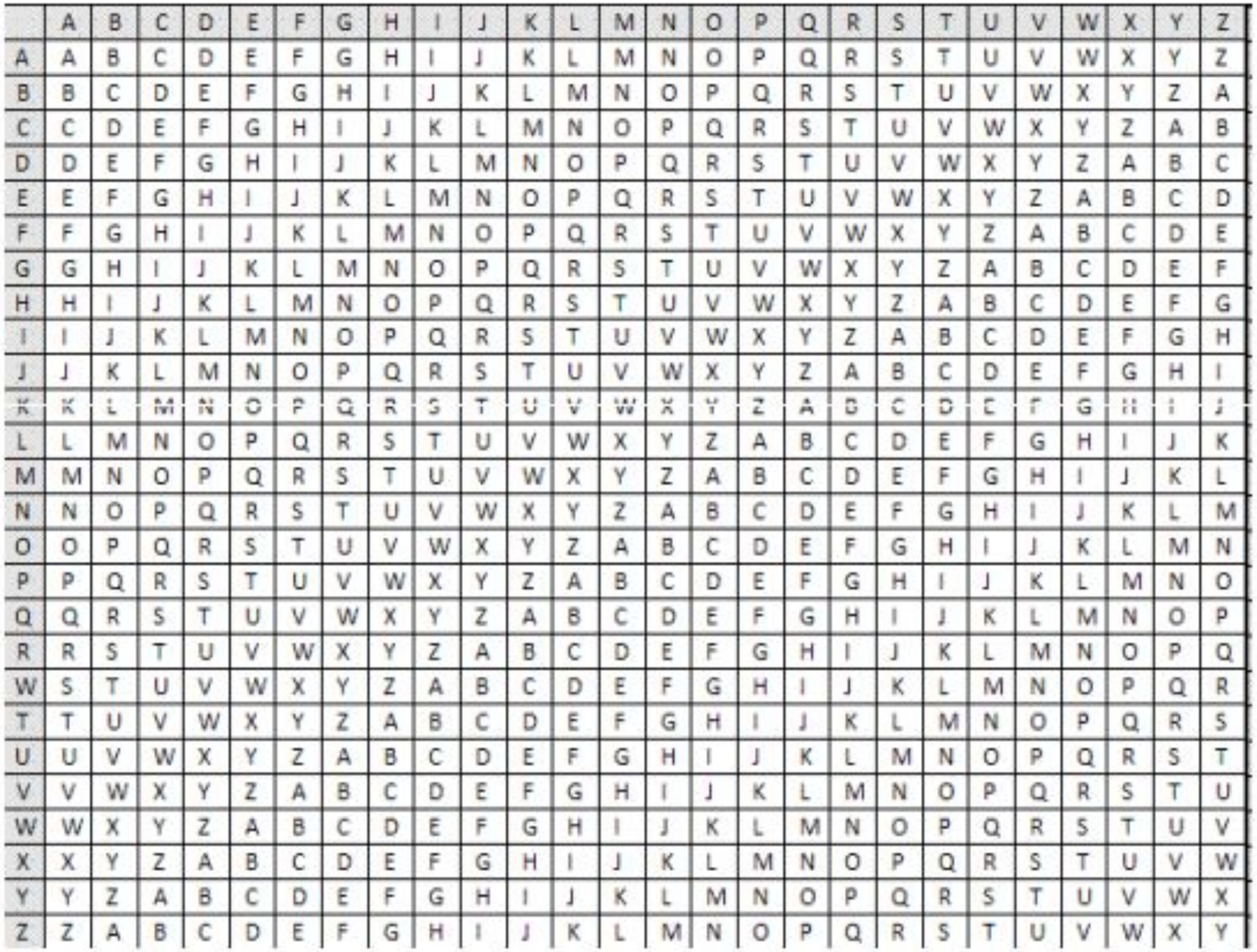
Example of Poly-alphabetic Cipher

- For key v & PT i , the corresponding CT is at the *intersection of row titled v & column titled i.*
- *For encryption*

  1. Key=PT

  2. Key repeats itself after n period.

Blaise de Vigenere (1523-1596) – *French diplomat and cryptographer – See next slide*



	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
W	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

Play Fair Square

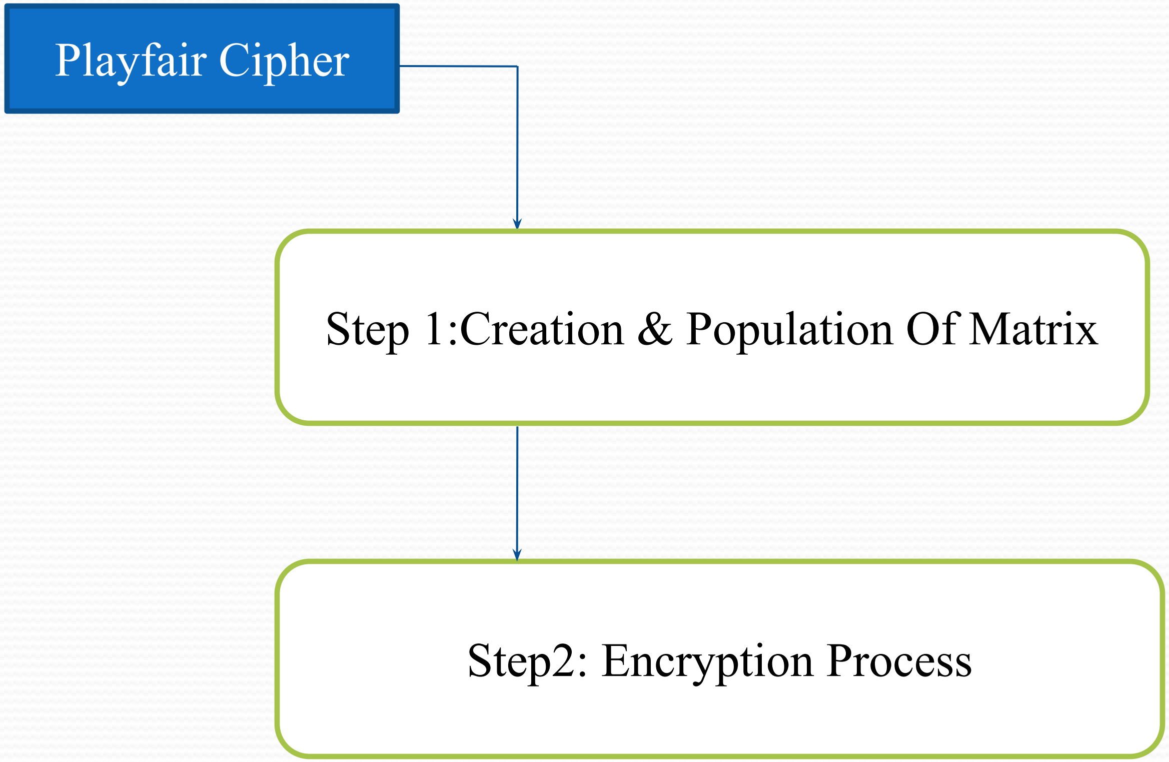
- Used for Manual Encryption Of Data
- Invented by Charles Wheatstone in 1854
- But known by Lord Playfair, who was friend of CW
- Playfair made this scheme popular & hence his name was used.
- Used by British Army in World War I
- Australians in World War II

Use

- Fast
- No equipments used.
- Important but not critical information.
- At the one could break it ->Value of information was nullified.
- *Crosswords that appears in several Newspapers.*

Playfair Encryption Scheme

- Uses to main processes


Playfair Cipher
Step 1:Creation & Population Of Matrix
Step2: Encryption Process

Step 1:Creation & Population Of Matrix

- 5*5 Matrix table
- Used to store keyword & phrase that becomes Key for E,D.

1. Enter the keyword in the matrix row-wise: L->R Top-to-bottom.
2. Drop duplicate letters.
3. Fill the remaining spaces in the matrix with the rest of the English alphabets(A-Z) that was not the part of our keyword.

Note:-While doing so, combine I & J in the same cell of the table. If I & J is the part of keyword, disregard both I & J while filling the remaining slots.

Step2: Encryption Process

1. **PT** broken into groups of two alphabets.
2. If both the alphabets are same OR *only one is left, add an X after the first alphabet.* **E** *the new pair & continue.*
3. If both the alphabets in the pair appear in the *same row* of our matrix, replace them with the alphabets of their *immediate right* respectively. If the original pair is on the right side of the row, then wrapping around to the *left side of the row* happens.
4. If both the alphabets in the pair appear in the *same coloum* of our matrix, replace them with the alphabets of their *immediate below* respectively. If the original pair is on the bottom side of the row, then wrapping around to the *top side of the row* happens.

Continue…

5. If the alphabets are not in *same row or column* of our matrix, replace them with the alphabets in the *same row* respectively , but the other pair of the corners of the rectangle defined by the original pair, The order is quite significant here. The first **E** alphabet of the pair is the one that is present on the same row as the first plain text alphabet.

- **D process works in the opposite direction. We also need to remove the extra X** alphabets that we had added in step 1# above, if any.

Hill Cipher

- Works on multiple letters at the same time(i.e: Polygraphic).
- Lester Hill invented this in 1929.
- Its roots in Matrix Theory of Mathematics.
- Inverse of Matrix
- Attack:- Known Plain Text Attack

Encryption Steps

1. Treat every letter in **PT** message as a number, so that **A=0,B=1,…….,Z=25.**
2. The **PT** matrix organized as a number.
3. **PT** is multiplied by a *randomly chosen keys*. The key matrix consist of size n*n, where n is the number of **rows** in **PT**.
4. Multiply the two matrix.
5. Now compute a mod26 value of the above matrix.
6. Now, translate the numbers to alphabets.

Decryption Steps

1. CT matrix multiply it by the Inverse of original key Matrix.
2. Take mod26 of the above matrix.
3. Now, translate the numbers to alphabets

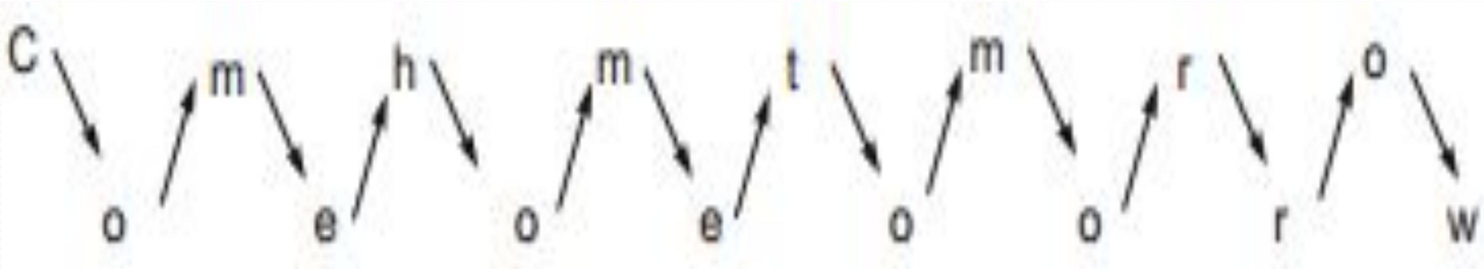
Transposition Ciphers

- A transposition cipher is one that permutes the symbols of the message according to a preset pattern.
  - Helps avoid detection of symbols based on correspondence.
  - Said to increase *diffusion*
    - Reduce redundancies in plaintext.

Rail Fence Technique

1. Write down the plain text message as a sequence of diagonals(*i.e: zigzag sequence*).
2. Read the text row-by-row, & write it sequentially.

- ✓ Original plain text message: **Come home tomorrow**



- ✓ we have the cipher text as: **Cmhmtmrooeoeoorw**
- ✓ Simple too break

Simple Columnar Transposition Technique

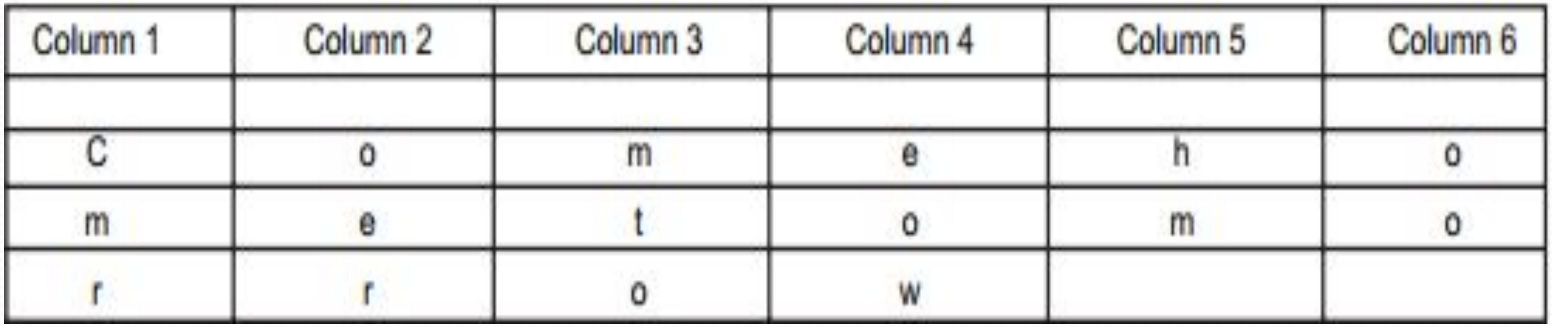
- *It simply arranges the plain text as a sequence of rows of a rectangle that are read in columns randomly.*

1. Write the plain text message row-by-row in a rectangle of a pre-defined size.
2. Read the message column-by-column. However, it need not be in the order of columns 1,2, 3 etc. It can be any random order such as 2, 3, 1, etc.
3. The message thus obtained is the cipher text message.

- Trying out a few permutations and combinations of column orders quite simple to break into.

- Original plain text message: **Come home tomorrow**

1. Let us consider a rectangle with *six columns*. Therefore, when we write the message in the rectangle row-by-row(suppressing spaces), it would look as follows



Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
C	o	m	e	h	o
m	e	t	o	m	o
r	r	o	w		

2. Now, let us decide the order of columns as some random order, say 4, 6, 1, 2, 5, 3. Then read the text in the order of these columns.
3. we have the cipher text as: **eowoocmroerhmmto**.

Simple columnar transposition technique with multiple rounds

- To add more complexity & twist using no. of iterations.
- Only one additional step is added is given below
- Repeat step 1 to 3 as many times as desired.

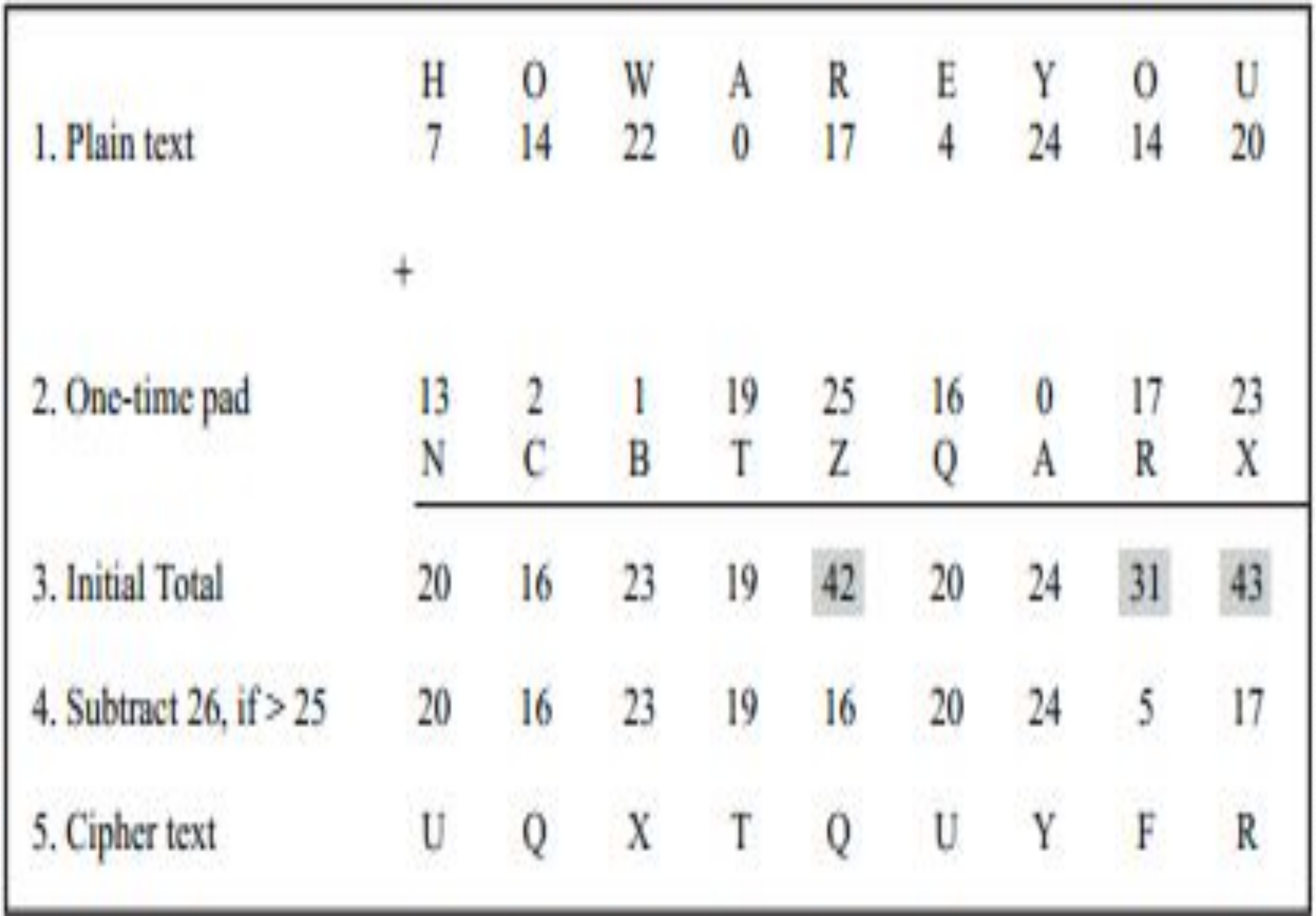
Vernam Cipher

- In 1920's was first implemented at AT&T with the help of a device called as the Vernam Machine.
- once an input cipher text for transposition is used, it is never used again for any other message (hence the name one-time pad).
- Random set of non-repeating characters as the input cipher text key.
  - Same length as message
  - XORed with message
- Theoretically unbreakable
  - Attacker can do no better than guessing
  - Ciphertext gives no information about plaintext.

Vernam Cipher Algorithm

1. Treat each plain text alphabet as a number in an increasing sequence, i.e. A = 0, B = 1, …Z = 25.
2. Do the same for each character of the input cipher text.
3. each number corresponding to the plain text alphabet to the corresponding input cipher text alphabet number.
4. If the sum thus produced is greater than 26, subtract 26 from it.
5. Translate each number of the sum back to the corresponding alphabet. This gives the output cipher text.

Example of Vernam Cipher



	H	O	W	A	R	E	Y	O	U
1. Plain text	7	14	22	0	17	4	24	14	20
+									
2. One-time pad	13	2	1	19	25	16	0	17	23
	N	C	B	T	Z	Q	A	R	X
3. Initial Total	20	16	23	19	42	20	24	31	43
4. Subtract 26, if > 25	20	16	23	19	16	20	24	5	17
5. Cipher text	U	Q	X	T	Q	U	Y	F	R

Book Cipher/Running Key Cipher

- For producing cipher text, *some portion of text from a book is used*, which serves the purpose of a one-time pad.
- Thus, the characters from a book are used as one-time pad, and they are *added* to the input plain text message similar to the way a one-time pad works.
- It should be clear that since the one-time pad is discarded after a single use.
- This technique is highly secure and suitable for small plain text message, but is clearly impractical for largemessages.

Product ciphers

- By themselves, substitution and transposition ciphers are relatively insecure.
- By combining these operations, we can produce a secure cipher.
  - This is how DES works.
- M -> Sub(M) -> Trans(Sub(M)).
  - Might go through multiple rounds.

Symmetric Key Encryption

- The Caesar Cipher and the one-time pad are examples of symmetric-key (secret-key) encryption.
- Single key shared by all users.
- Fast
- How to distribute keys?

Keyspace

- The *keyspace* is the set of all possible keys.
  - Caesar cipher: keyspace = {0,1,2,…,25}
  - Vernam cipher: |keyspace| = 2^n –1

- Size of the keyspace helps us estimate security.
  - Assumption: exhaustive search is the only way to find a key.

Block Ciphers

- The ciphers we have seen so far are known as *block ciphers*.
- Plaintext is broken into blocks of size *k*.
- Each block is encrypted separately.
- Advantages: random access, potentially high security
- Disadvantages: larger block size needed, patterns retained throughout messages.

Stream Ciphers

- A stream cipher encodes a symbol based on both the key and the encoding of previous symbols.
  - $C_i = M_i \text{ XOR } K_i \text{ XOR } M_{i-1}$
- Advantages:
  - can work on smaller block sizes – little memory/processing/buffering needed.
- Disadvantages:
  - Random access difficult, hard to use large keys.
  - Sender and receiver must be synchronized
    - Inserted bits can lead to errors.