Cryptography

Section 3

Created by/ Rehab Mohamed & Hanaa Mansour

Vigenere Cipher

- Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution.
- Vigenere Cipher will use a letter key instead of a numeric key representation.

Encryption using Vigenere Cipher

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

$$c = (p + k) \bmod 26$$

- Example: Encrypt the message "HELLO THERE", given the key = "ITEAM"

H	E	L	L	O		T	H	E	R	E
I	T	E	A	M		I	T	E	A	M

H: 7 + 8 = 15 mod 26 = 15 : P
E: 4 + 19 = 23 mod 26 = 23: X
L: 11 + 4 = 15 mod 26 = 15 : P
L: 11 + 0 = 11 mod 26 = 11 : L
O: 14 + 12 = 26 mod 26 = 0 : A

PXPLA

T: 19 + 8 = 27 mod 26 = 1 : B
H: 7 + 19 = 26 mod 26 = 0: A
E: 4 + 4 = 8 mod 26 = 8 : I
R: 17 + 0 = 17 mod 26 = 17 : R
E: 4 + 12 = 16 mod 26 = 16 : Q

BAIRQ

Decryption using Vigenere Cipher

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

$$c = (p - k) \bmod 26$$

- Decrypt the message "PXPLA BAIRQ", given the key = "ITEAM"

P	X	P	L	A		B	A	I	R	Q
I	T	E	A	M		I	T	E	A	M

P: 15 - 8 = 7 mod 26 = 7 : H

X: 23 - 19 = 4 mod 26 = 4: E

P: 15 - 4 = 11 mod 26 = 11 : L

L: 11 - 0 = 11 mod 26 = 11 : L

A: 0 - 12 = -12 + 26 = 14 mod 26 = 14 : O

HELLO

B: 1 - 8 = -7+26= 19 mod 26 = 19 : T

A: 0 - 19 = -19+26=7 mod 26 = 7: H

I: 8 - 4 = 4 mod 26 = 4 : E

R: 17 - 0 = 17 mod 26 = 17 : R

Q: 16 - 12 = 4 mod 26 = 4 : E

THERE

```
import math
# This function generates the key in a cyclic manner until it’s length
isn't equal to the length of original text

def generateKey(string, key):
    if len(string) <= len(key):
        croped_key = key[:len(string)]
    else:
        ceil = math.ceil(len(string) / len(key))
        new_key = []
        for i in range(0, ceil):
            for l in key:
                new_key.append(l)
        croped_key = new_key[:len(string)]
    return (croped_key)
```

```
# This function returns the encrypted text generated with the help of the key
def cipherText(string, key):
    cipher_text = []
    for i in range(len(string)):
        #converting in range 0-25
        x = (ord(string[i]) + ord(key[i])) % 26
        #convert into alphabets(ASCII)
        x += ord('A')
        cipher_text.append(chr(x))
    return ("".join(cipher_text))

# This function decrypts the encrypted text and returns the original text
def originalText(cipher_text, key):
    orig_text = []
    for i in range(len(cipher_text)):
        x = (ord(cipher_text[i]) - ord(key[i]) + 26) % 26
        x += ord('A')
        orig_text.append(chr(x))
    return ("".join(orig_text))
```

```
string = "HELLOTHERE"
keyword = "ITEAM"
key = generateKey(string, keyword)
cipher_text = cipherText(string, key)
print("Cipher text :", cipher_text)
print("Original/Decrypted Text :", originalText(cipher_text, key))
```

Output:

```
Cipher text : PXPLABAIRQ
Original/Decrypted Text : HELLOTHERE
```


Thank
you