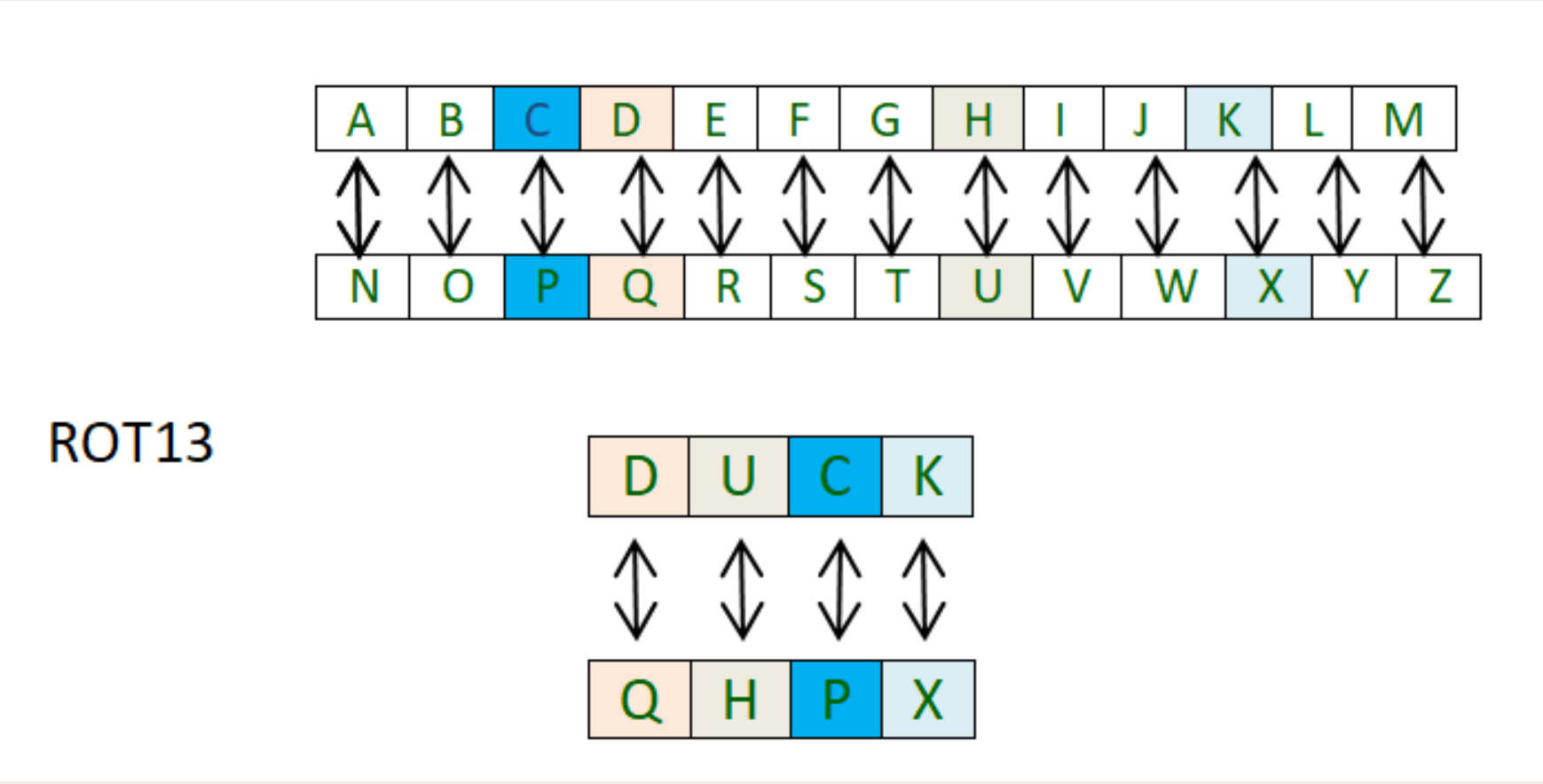
Cryptography

Section 4

Created by/ Rehab Mohamed & Hanaa Mansour

ROT13 cipher

- ROT13 cipher (read as – “rotate by 13 places”) is a special case of the Caesar cipher in which the shift is always 13. So, every letter is shifted 13 places to encrypt or to decrypt the message.



Encryption using ROT13 Cipher

```
def encrypt(text, k):
    result = ''
    for c in text:
        if (c != ' '):
            if (c.isupper()):
                # Encrypt uppercase characters
                s = chr((ord(c) - 65 + k) % 26 + 65)
            else:
                # Encrypt lowercase characters
                s = chr((ord(c) - 97 + k) % 26 + 97)
        else:
            s = ' '
        result += s
    return result
```

Decryption using ROT13 Cipher

```
def decrypt(cipher, k):
    result = ''
    for c in cipher:
        if (c != ' '):
            if (c.isupper()):
                # Encrypt uppercase characters
                s = chr((ord(c) - 65 - k) % 26 + 65)
            else:
                # Encrypt lowercase characters
                s = chr((ord(c) - 97 - k) % 26 + 97)
        else:
            s = ' '
        result += s
    return result
```

```
message = "Hello World"
key = 13
ciphertext = encrypt(message, key)
print("The cipher text is: " + ciphertext)
plaintext = decrypt(ciphertext,key)
print("The decrypted text is: " + plaintext)
```

Output:

```
The cipher text is: Uryyb Jbeyq
The decrypted text is: Hello World
```

Multiplicative Cipher

- While using Caesar cipher technique, encrypting and decrypting symbols involves converting the values into numbers with a simple basic procedure of addition or subtraction.
- If multiplication is used to convert to cipher text, it is called a **wrap-around** situation. Consider the letters and the associated numbers to be used as shown below.

$$C = E(P) = (P * K) \bmod n$$ ← قانون التشفير

$$P = D(C) = (C * k^{-1}) \bmod n$$ ← قانون فك التشفير

Note: The Greatest Common Divisor (GCD) between the key and n should equal 1.

Calculate GCD

Key = 9 , n = 26

```
import math
print(math.gcd(26,9))     #1
```

Example: **Encrypt** the word "**secret**" using **multiplicative** cipher with **key=9** ~~~~~~~ **n=26**

$C=(p*k) \bmod N$

Index	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
Alphabet	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

secret

plaintext	Encryption $C=(p\times k) \bmod 26$	Ciphertext
S=18	$C=(18\times 9) \bmod 26$	6=G
E=4	$C=(4\times 9) \bmod 26$	10=K
C=2	$C=(2\times 9) \bmod 26$	18=S
R=17	$C=(17\times 9) \bmod 26$	23=X
E=4	$C=(4\times 9) \bmod 26$	10=K
T=19	$C=(19\times 9) \bmod 26$	15=P
		GKSXKP

Encryption using Multiplicative Cipher

```
def encrypt(text, k):
    result = ''
    for c in text:
        if (c != ' '):
            if (c.isupper()):
                # Encrypt uppercase characters
                s = chr(((ord(c) - 65) * k) % 26 + 65)
            else:
                # Encrypt lowercase characters
                s = chr(((ord(c) - 97) * k) % 26 + 97)
        else:
            s = ' '
        result += s
    return result
```

```
message = "Secret"
key = 9
ciphertext = encrypt(message, key)
print("The cipher text is: " + ciphertext)
```

Output:

```
The cipher text is: Gksxkp
```


Thank
you