NAME: PRERNA SUNIL JADHAV
SAPID: 60004220127
BATCH: C2-2
BRANCH: COMPUTER ENGINEERING
COURSE: INFORMATION SECURITY LABORATORY
COURSE CODE: DJ19CEL603

EXPERIMENT 03

AIM: Study and implement vernam cipher.

THEORY: Vernam cipher is a method of encaypting alphabetic text. It is one of the substitution techniques for converting plain text into cipher text.
In this mechanism, we assign a number to each character of the plain text, like (a=0, b=1, c=2, ... z=25). Method to take key: In the vernam cipher algorithm, we take a key to encrypt the plain text whose length should be equal to the length of plain text.
Encuyption Algorithm:
1) Assign a number to each character of the plain text and key according to the alphabetic order.
2) Bitwise XOR both the numbers (corresponding plain-text character number and key character number).

3) Subtract the number from 26 if the resulting number is greater than or equal to 26, if it isnt then leave it.

Eg: Plaintext: "OAK"
Key : "SON"

Key:	S	O	N
	18	14	13
Plaintext:	O	A	K
	18	00	10
Ciphertext:	C	O	H
	02	14	07

∴ ciphertext: "COH"

CONCLUSION: Both encryption and decryption algorithm are simple and involve a bitwise XOR operation. This simplicity can be an advantage in some situation.
But the key must be at least as long as the message, which can be inefficient for long messages.
Thus, we studied and implemented vernam cipher.

Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)

Academic Year: 2022-2023

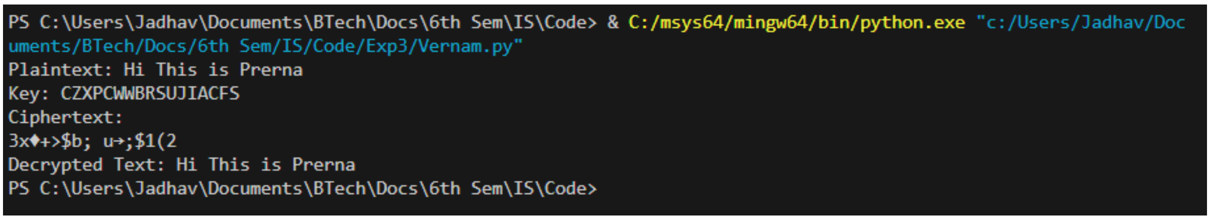
Name:	Prerna Sunil Jadhav
Sap Id:	60004220127
Class:	T. Y. B. Tech (Computer Engineering)
Course:	Information Security Laboratory
Course Code:	DJ19CEL603
Experiment No.:	03

AIM: Study and Implement Vernam Cipher.

CODE:

```
import random
def generate_key(plaintext_length):
    key = ''.join(random.choice('ABCDEFGHIJKLMNOPQRSTUVWXYZ') for _ in
range(plaintext_length))
    return key

def encrypt(plaintext, key):
    ciphertext = ''.join(chr(ord(p) ^ ord(k)) for p, k in zip(plaintext, key))
    return ciphertext
def decrypt(ciphertext, key):
    decrypted_text = ''.join(chr(ord(c) ^ ord(k)) for c, k in zip(ciphertext,
key))
    return decrypted_text

if __name__ == "__main__":
    plaintext = "Hi This is Prerna"
    key = generate_key(len(plaintext))

    print("Plaintext:", plaintext)
    print("Key:", key)

    ciphertext = encrypt(plaintext, key)
    print("Ciphertext:", ciphertext)

    decrypted_text = decrypt(ciphertext, key)
    print("Decrypted Text:", decrypted_text)
```

OUTPUT:



```
PS C:\Users\Jadhav\Documents\BTech\Docs\6th Sem\IS\Code> & C:/msys64/mingw64/bin/python.exe "c:/Users/Jadhav/Doc
uments/BTech/Docs/6th Sem/IS/Code/Exp3/Vernam.py"
Plaintext: Hi This is Prerna
Key: CZXPCWWBRSUJIACFS
Ciphertext:
3x♦+>$b; u→;$1(2
Decrypted Text: Hi This is Prerna
PS C:\Users\Jadhav\Documents\BTech\Docs\6th Sem\IS\Code>
```