Cryptography and Network Security Lab

Assignment 4

Student Details

Name : Krunal Rank
Adm No. : U18CO081

1

```
# sample text : once upon a time there was a little girl named goldilocks she went
for a walk in the forest pretty soon she came upon a house she knocked and when no
one answered she walked right in at the table in the kitchen there were three bowls
of porridge goldilocks was hungry she tasted the porridge from the first bowl
import time

class ERRORS:
    """
    Error Messages
    """
    INVALID_CHOICE = "Please enter a valid Integer choice."
    INVALID_TEXT = "Please enter a valid Text. Text must only contain Lowercase
Alphabets."

FILE_NAME = "encrypted_text.txt"

def vernam_cipher_encrypt(text:str):
    """
    Vernam Cipher Encryption using Key generated from timestamp. Timestamp is also
returned to the User.
    Returns:
        Encrypted Text
        Key
    """
    key = int(time.time())

    text = text.lower()

    for c in text:
        if not (c.isalpha() or c.isspace()):
            raise Exception(ERRORS.INVALID_TEXT)

    encrypted_text = ""
```

```
    for i in range(len(text)):
        encrypted_text += chr(ord(text[i]) ^ (key * (i+1))%128)

    return encrypted_text, key

def vernam_cipher_decrypt(text:str, key:int):
    """
    Vernam Cipher Decryption using given Key.
    Returns:
        Decrypted Text
    """

    decrypted_text = ""
    for i in range(len(text)):
        decrypted_text += chr(ord(text[i]) ^ (key * (i+1))%128)

    return decrypted_text


def vernam_cipher_encrypt_dialog():
    """
    Runs Vernam Cipher Encryption Dialog
    """
    text = input("Enter text to be encrypted: ")

    encrypted_text,key = vernam_cipher_encrypt(text)

    # Save text in file in case it is malformed in the terminal
    # Note that the key won't be saved in the file. It needs to be secured by the
User.
    with open(FILE_NAME,"wb") as f:
        f.write(bytes(encrypted_text, "utf-8"))

    print(f"Encrypted Text: {encrypted_text}\nKey (Do not share) : {key}")


def vernam_cipher_decrypt_dialog():
    """
    Runs Vernam Cipher Decryption Dialog
    """
    text = input("Enter text to be decrypted: ")
    key = input("Enter key: ")

    decrypted_text = vernam_cipher_decrypt(text, key)

    print(f"Decrypted Text: {decrypted_text}")
```

```
def vernam_cipher_file_decrypt_dialog():
    """
    Runs Vernam Cipher Decryption Dialog where text is obtained from file.
    """
    key = int(input("Enter key: "))
    with open(FILE_NAME,"rb") as f:
        text = f.read().decode("utf-8")
    decrypted_text = vernam_cipher_decrypt(text, key)

    print(f"Decrypted Text: {decrypted_text}")

def main_dialog():
    """
    Runs main dialog sequence
    """
    try:
        choice = int(input(
            "Vernam Cipher (One Time Pad) Program\n1. Encrypt\n2. Decrypt\n3. Decrypt
from encrypted_text.txt\nPlease enter your choice: "))
    except Exception as e:
        raise Exception(ERRORS.INVALID_CHOICE)

    if choice == 1:
        vernam_cipher_encrypt_dialog()
    elif choice == 2:
        vernam_cipher_decrypt_dialog()
    elif choice == 3:
        vernam_cipher_file_decrypt_dialog()
    else:
        raise Exception(ERRORS.INVALID_CHOICE)


if __name__ == "__main__":
    try:
        main_dialog()
    except Exception as e:
        print(e)
```

```
kr@arc-warden:/mnt/6AD574E142A88B4D/BTech/Assignments/4th_Year/CNS/Assignment_4$ python3 1.py
Vernam Cipher (One Time Pad) Program
1. Encrypt
2. Decrypt
3. Decrypt from encrypted_text.txt
Please enter your choice: 1
Enter text to be encrypted: hi i am krunal
Encrypted Text: (i`i`a- +r5n!l
Key (Do not share) : 1630494272
kr@arc-warden:/mnt/6AD574E142A88B4D/BTech/Assignments/4th_Year/CNS/Assignment_4$ python3 1.py
Vernam Cipher (One Time Pad) Program
1. Encrypt
2. Decrypt
3. Decrypt from encrypted_text.txt
Please enter your choice: 3
Enter key: 1630494272
Decrypted Text: hi i am krunal
kr@arc-warden:/mnt/6AD574E142A88B4D/BTech/Assignments/4th_Year/CNS/Assignment_4$
```