ಜಿ. ಎಂ. ವಿಶ್ವವಿದ್ಯಾಲಯ
GM UNIVERSITY
P. B. Road, Davanagere – 577 006 KARNATAKA | INDIA

Faculty of Computing and IT

Master in Computer Applications

Class:1st SEM

Course: Cyber Security

Assigment-1

Implementation Of Cryptography Using Random Encryption

Code

```
import base64

key = "0123456789"

def xor_crypt(msg, key):
    return base64.urlsafe_b64encode("".join(chr(ord(m) ^ ord(key[i % len(key)])) for i, m in
enumerate(msg)).encode()).decode()

def xor_decrypt(enc, key):
    return "".join(chr(ord(m) ^ ord(key[i % len(key)])) for i, m in
enumerate(base64.urlsafe_b64decode(enc).decode()))

msg = input("Enter your message: ")
enc_msg = xor_crypt(msg, key)
print("The Encrypted message is:", enc_msg)

dec_msg = xor_decrypt(enc_msg, key)
print("The Decrypted message is:", dec_msg)
```

output

```
C:\Users\sande\PycharmProjects\Stegnography\.venv\Scripts\python.exe C:\Users\sande\PycharmProjects\Stegnography\crypto.py
Enter your message: sandeep
The Encrypted message is: Q1BcV1FQRg==
The Decrypted message is: sandeep

Process finished with exit code 0
```

Assignment-2

Random Encryption in Server Hack

Server Hack A server hack refers to unauthorized access, manipulation, or exploitation of a server. This can be done using various techniques, including exploiting vulnerabilities, misconfigurations, or weak credentials.

Sender Code

```
import socket

def encrypt_caesar(text, shift=3):
    result = ""
    for char in text:
        if char.isalpha():
            shift_base = ord('A') if char.isupper() else ord('a')
            result += chr((ord(char) - shift_base + shift) % 26 + shift_base)
        else:
            result += char
    return result


message = input("Enter your message: ")
encrypted_message = encrypt_caesar(message)


server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((socket.gethostname(), 4000))
server.listen(5)
print("Server is running and waiting for connections...")

while True:
    client, address = server.accept()
    print(f"Connection from {address} established.")
```

```
client.send(bytes(encrypted_message, "utf-8"))
client.close()
```

Output:-

```
C:\Users\sande\PycharmProjects\Stegnography\.venv\Scripts\python.exe C:\Users\sande\PycharmProjects\Stegnography\send.py
Enter your message: sandeep
Server is running and waiting for connections...
Connection from ('192.168.26.47', 55050) established.
```

Receiver Code

```
import socket

def decrypt_caesar(text, shift=3):
    result = ""
    for char in text:
        if char.isalpha():
            shift_base = ord('A') if char.isupper() else ord('a')
            result += chr((ord(char) - shift_base - shift) % 26 + shift_base)
        else:
            result += char
    return result

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((socket.gethostname(), 4000))

encrypted_message = client.recv(1024).decode("utf-8")
print("Encrypted Message from Server:", encrypted_message)

key = int(input("Enter the decryption key: "))

original_message = decrypt_caesar(encrypted_message, key)
print("Decrypted Message:", original_message)

client.close()
```

Output:-

```
C:\Users\sande\PycharmProjects\Stegnography\.venv\Scripts\python.exe C:\Users\sande\PycharmProjects\Stegnography\recive.py
Encrypted Message from Server: vdqghhs
Enter the decryption key: 3
Decrypted Message: sandeep

Process finished with exit code 0
```

Hacker Code

```
import socket

def decrypt_caesar(text, shift):
    result = ""
    for char in text:
        if char.isalpha():
            shift_base = ord('A') if char.isupper() else ord('a')
            result += chr((ord(char) - shift_base - shift) % 26 + shift_base)
        else:
            result += char
    return result


s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((socket.gethostname(), 4000))


encrypted_message = s.recv(1024).decode("utf-8")
print("Intercepted Encrypted Message:", encrypted_message)


print("\nAttempting to brute-force the encryption...\n")

for key in range(26):
    decrypted_text = decrypt_caesar(encrypted_message, key)
    print(f"Key {key}: {decrypted_text}")

s.close()
```

Output:-

```
C:\Users\sande\PycharmProjects\Stegnography\.venv\Scripts\python.exe C:\Users\sande\PycharmProjects\Stegnography\Hackerrr.py
Intercepted Encrypted Message: vdqghhs

Attempting to brute-force the encryption...

Key 0: vdqghhs
Key 1: ucpfggr
Key 2: tboeffq
```

Assignment-3

3. Secure Password Generator : A Secure Password Generator is a tool that creates strong, random passwords that are difficult for hackers to guess or crack. These passwords typically include a mix of uppercase and lowercase letters, numbers, and special characters to enhance security.

Code

```
import secrets
import string


def generate_password(length=16):
    if length < 8:
        raise ValueError("Password length must be at least 8 characters for security.")


    all_chars = string.ascii_uppercase + string.ascii_lowercase + string.digits +
string.punctuation


    password = ''.join(secrets.choice(all_chars) for _ in range(length))

    return password


secure_password = generate_password(16)
print("Generated Secure Password:", secure_password)
```

output:-

```
C:\Users\sande\PycharmProjects\Stegnography\.venv\Scripts\python.exe C:\Users\sande\PycharmProjects\Stegnography\password.py
Generated Secure Password: B:g(ZMA@E4Ba/\W|

Process finished with exit code 0
```

Assignment-4

Demonstrate code for Sentiment Analysisnt

Sentiment Code

```
from textblob import TextBlob


def analyze_sentiment(text):
    blob = TextBlob(text)
    polarity = blob.sentiment.polarity

    if polarity > 0:
        return "Positive"
    elif polarity < 0:
        return "Negative"
    else:
        return "Neutral"

if __name__ == "__main__":
    text = input("Enter a sentence: ")
    sentiment = analyze_sentiment(text)
    print(f"Sentiment: {sentiment}")
```

output:-

```
C:\Users\sande\PycharmProjects\Sentiment\.venv\Scripts\python.exe C:\Users\sande\PycharmProjects\Sentiment\sentii.py
Enter a sentence: i love you
Sentiment: Positive

Process finished with exit code 0
```