Cyber Security Fundamentals

Assignment 1

Python script used in VS Code to encrypt and decrypt using the Caesar cipher with the specified shift:

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

def caesar_decrypt(text, shift):
    return caesar_encrypt(text, -shift)

def main():
    shift = 4

    # Prompt user for message to encrypt
    message = input("Enter message to encrypt: ")
    encrypted = caesar_encrypt(message, shift)
    print("Encrypted message:", encrypted)

    # Prompt user for message to decrypt
    ciphertext = input("Enter message to decrypt: ")
    decrypted = caesar_decrypt(ciphertext, shift)
    print("Decrypted message:", decrypted)

if __name__ == "__main__":
    main()
```

1. Define the Encryption Function

The **caesar_encrypt** function shifts each alphabetic character in the input text forward by the specified shift.

- It loops through each character.
- Checks if the character is a letter (ignores punctuation and spaces).
- Preserves case (uppercase/lowercase) using ASCII values.
- Converts characters with wrapping around Z to A.
- Non-alpha characters are added unchanged.

Example: 'A' shifted by 4 becomes 'E'.

2. Define the Decryption Function

The **caesar_decrypt** function simply calls **caesar_encrypt** with the negative of the shift:

- To decrypt, we move letters backward by the same shift.
- This reuse of the encrypt function keeps the code concise and clear.

3. Main Program Logic

The **main()** function runs when you execute the script.

- Sets the shift to 4.
- Prompts the user to enter a message to encrypt.
- Prints the encrypted message.
- Prompts the user to enter a ciphertext to decrypt.
- Prints the decrypted message.

This interaction happens in the terminal with simple input/output.

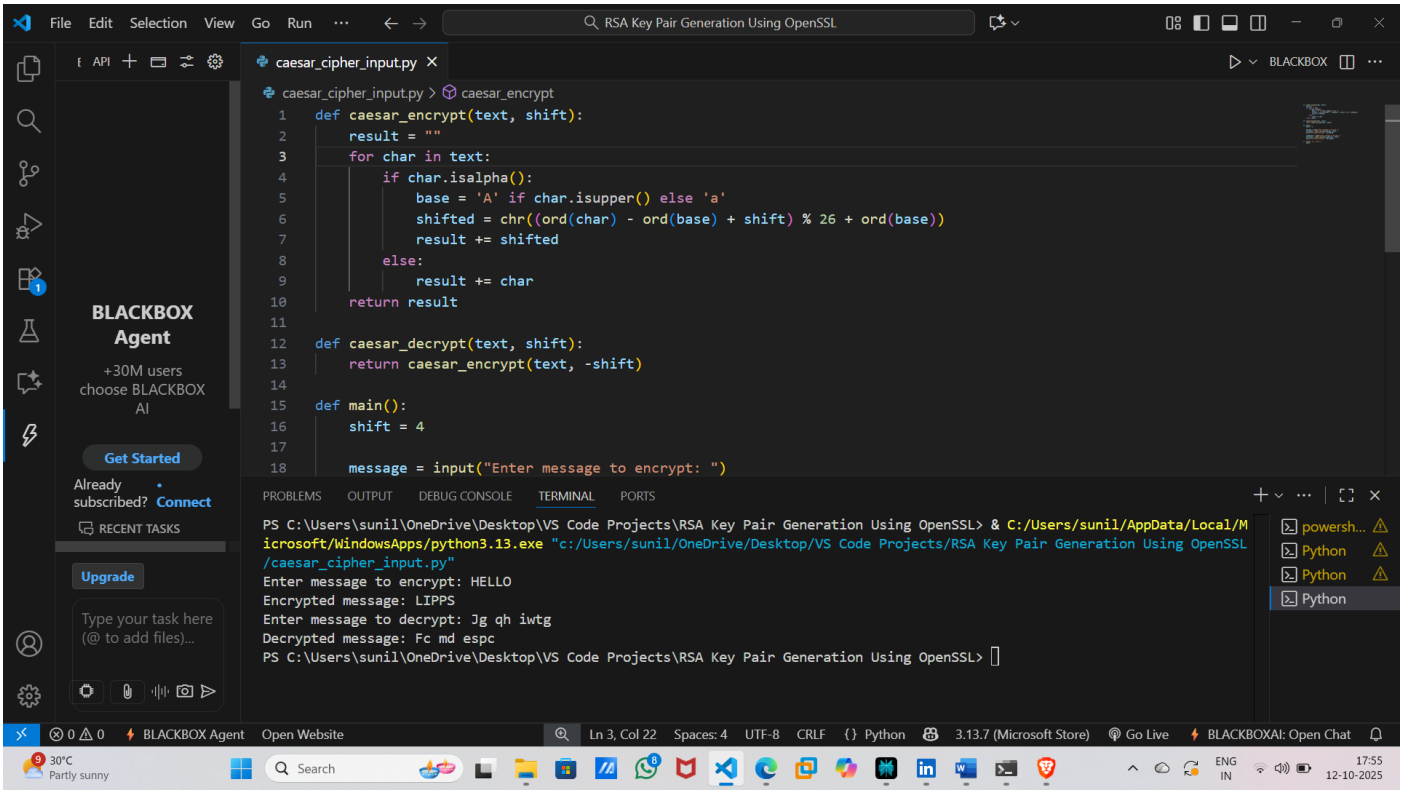
4. Running the Script

- Open VS Code terminal.
- Execute with python **caesar_cipher_input.py**.
- Follow prompts to enter your messages.

This makes the program easy to test with different inputs without changing the code.

Key Takeaways

- The Caesar cipher shifts letters with wrapping.
- Case and non-letter characters are handled gracefully.
- The script is interactive, user-friendly.
- Code reuse by calling encryption with negative shift for decryption.


File Edit Selection View Go Run
RSA Key Pair Generation Using OpenSSL
caesar_cipher_input.py
BLACKBOX
caesar_cipher_input.py > caesar_encrypt
def caesar_encrypt(text, shift):
result = ""
for char in text:
if char.isalpha():
base = 'A' if char.isupper() else 'a'
shifted = chr((ord(char) - ord(base) + shift) % 26 + ord(base))
result += shifted
else:
result += char
return result
def caesar_decrypt(text, shift):
return caesar_encrypt(text, -shift)
def main():
shift = 4
message = input("Enter message to encrypt: ")
BLACKBOX Agent
+30M users choose BLACKBOX AI
Get Started
Already subscribed? Connect
RECENT TASKS
Upgrade
Type your task here (@ to add files)...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\sunil\OneDrive\Desktop\VS Code Projects\RSA Key Pair Generation Using OpenSSL> & C:/Users/sunil/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/sunil/OneDrive/Desktop/VS Code Projects/RSA Key Pair Generation Using OpenSSL/caesar_cipher_input.py"
Enter message to encrypt: HELLO
Encrypted message: LIPPS
Enter message to decrypt: Jg qh iwtg
Decrypted message: Fc md espc
PS C:\Users\sunil\OneDrive\Desktop\VS Code Projects\RSA Key Pair Generation Using OpenSSL>
powersh...
Python
Python
Python
BLACKBOX Agent
Open Website
Ln 3, Col 22
Spaces: 4
UTF-8
CRLF
Python
3.13.7 (Microsoft Store)
Go Live
BLACKBOXAI: Open Chat
30°C
Partly sunny
Search
ENG
IN
17:55
12-10-2025

Assignment 2: Question: Use the openssl command-line tool to generate an RSA key pair (public and private key). Export the keys and display them in PEM format.

Step 1: Open Your Terminal or Command Prompt

Make sure OpenSSL is installed and accessible. Type:

openssl version

You should see the OpenSSL version displayed.

Step 2: Generate the Private Key

Run the following command to generate a 2048-bit RSA private key and save it in private_key.pem:

openssl genrsa -out private_key.pem 2048

Step 3: Generate the Public Key from the Private Key

Create the public key public_key.pem with this command:

openssl rsa -in private_key.pem -pubout -out public_key.pem

Step 4: Verify the Private Key (View the PEM Format)

Display the private key file contents:

cat private_key.pem

Your output will begin with:

-----BEGIN RSA PRIVATE KEY-----

Step 5: Verify the Public Key (View the PEM Format)

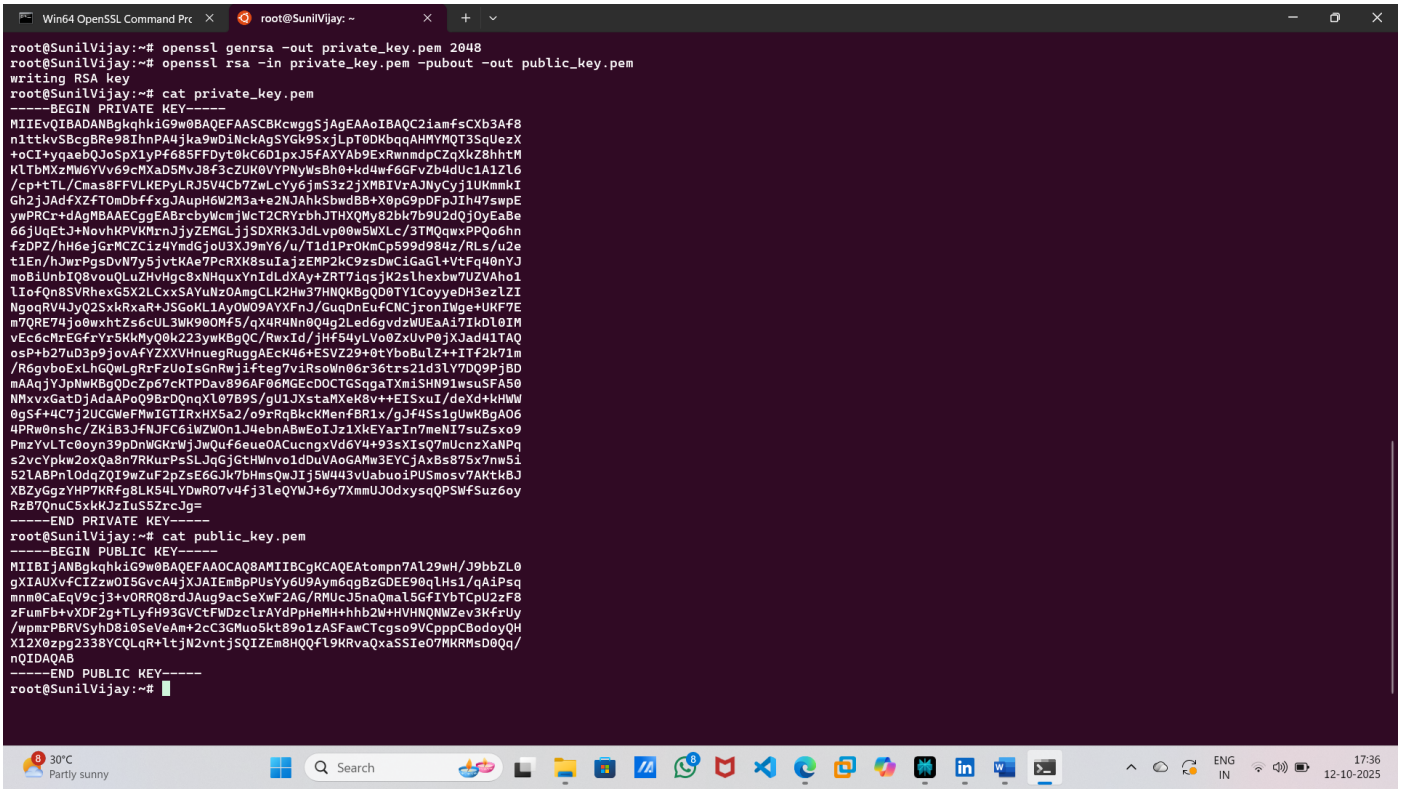
Display the public key file contents:

cat public_key.pem

Your output will begin with:

text

-----BEGIN PUBLIC KEY-----


Win64 OpenSSL Command Prc
root@SunilVijay: ~
root@SunilVijay:~# openssl genrsa -out private_key.pem 2048
root@SunilVijay:~# openssl rsa -in private_key.pem -pubout -out public_key.pem
writing RSA key
root@SunilVijay:~# cat private_key.pem
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC2iamfsCXb3Af8
n1ttkvSBcgBRe98IhnPA4jka9wDiNckAgSYGk9SxjLpT0DKbqqAHMYMQT3SqUezX
+oCI+yqaebQJoSpX1yPf685FFDyt0kC6D1pxJ5fAXYAb9ExRwnmdpCZqXkZ8hhtM
KlTbMXzMW6YVv69cMXaD5MvJ8f3czZUK0VYPNyWsBh0+kd4wf6GFvZb4dUc1A1Zl6
/cp+tTL/Cmas8FFVLKEPyLRJ5V4Cb7ZwLcYy6jmS3z2jXMBIVrAJNyCyj1UKmmkI
Gh2jJAdfXZfTOmDbffxgJAupH6W2M3a+e2NJAhkSbwdBB+X0pG9pDFpJIh47swpE
ywPRCr+dAgMBAAECggEABrcbyWcmjWcT2CRYrbhJTHXQMy82bk7b9U2dQjOyEaBe
66jUqEtJ+NovhKPVKMrnJjyZEMGLjjSDXRK3JdLvp00w5WXLc/3TMQqwxPPQo6hn
fzDPZ/hH6ejGrMCZCiz4YmdGjoU3XJ9mY6/u/T1d1PrOKmCp599d984z/RLs/u2e
t1En/hJwrPgsDvN7y5jvtKAe7PcRXK8suIajzEMP2kC9zsDwCiGaGl+VtFq40nYJ
moBiUnbIQ8vouQLuZHvHgc8xNHquxYnIdLdXAy+ZRT7iqsjK2slhexbw7UZVAho1
lIofQn8SVRhexG5X2LCxxSAYuNzOAmgCLK2Hw37HNQKBgQD0TY1CoyyeDH3ezlZI
NgoqRV4JyQ2SxkRxaR+JSGoKL1AyOWO9AYXFnJ/GuqDnEufCNCjronIWge+UKF7E
m7QRE74jo0wxhtZs6cUL3WK90OMf5/qX4R4Nn0Q4g2Led6gvdzWUEaAi7IkDl0IM
vEc6cMrEGfrYr5KkMyQ0k223ywKBgQC/RwxId/jHf54yLVo0ZxUvP0jXJad41TAQ
osP+b27uD3p9jovAfYZXVHnuegRuggAEcK46+ESVZ29+0tYboBulZ++ITf2k71m
/R6gvboExLhGQwLgRrFzUoIsGnRwjifteg7viRsoWn06r36trs21d3lY7DQ9PjBD
mAAqjYJpNwKBgQDcZp67cKTPDav896AF06MGEcDOCTGSqgaTXmiSHN91wsuSFA50
NMxvxGatDjAdaAPoQ9BrDQnqXl07B9S/gU1JXstaMXeK8v++EISxuI/deXd+kHWW
0gSf+4C7j2UCGWeFMwIGTIRxHX5a2/o9rRqBkcKMenfBR1x/gJf4Ss1gUwKBgAO6
4PRw0nshc/ZKiB3JfNJFC6iWZWOn1J4ebnABwEoIJz1XkEYarIn7meNI7suZsxo9
PmzYvLTc0oyn39pDnWGKrWjJwQuf6eueOACucngxVd6Y4+93sXIsQ7mUcnzXaNPq
s2vcYpkw2oxQa8n7RKurPsSLJqGjGtHWnvo1dDuVAoGAMw3EYCjAxBs875x7nw5i
52lABPnlOdqZQI9wZuF2pZsE6GJk7bHmsQwJIj5W443vUabuoiPUSmosv7AKtkBJ
XBZyGgzYHP7KRfg8LK54LYDwRO7v4fj3leQYWJ+6y7XmmUJOdxysqQPSWfSuz6oy
RzB7QnuC5xkKJzIuS5ZrcJg=
-----END PRIVATE KEY-----
root@SunilVijay:~# cat public_key.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtompn7Al29wH/J9bbZL0
gXIAUXvfCIZzwOI5GvcA4jXJAIEmBpPUsYy6U9Aym6qgBzGDEE90qlHs1/qAiPsq
mnm0CaEqV9cj3+vORRQ8rdJAug9acSeXwF2AG/RMUcJ5naQmal5GfIYbTCpU2zF8
zFumFb+vXDF2g+TLyfH93GVCtFWDzclrAYdPpHeMH+hhb2W+HVHNQNWZev3KfrUy
/wpmrPBRVSyhD8i0SeVeAm+2cC3GMuo5kt89o1zASFawCTcgso9VCpppCBodoyQH
X12X0zpg2338YCQLqR+ltjN2vntjSQIZEm8HQQfl9KRvaQxaSSIeO7MKRMsD0Qq/
nQIDAQAB
-----END PUBLIC KEY-----
root@SunilVijay:~#
30°C
Partly sunny
Search
ENG
IN
17:36
12-10-2025