

10. Python program implementation asymmetric encryption using rsa python library.

```
import rsa

def generate_keys():
    return rsa.newkeys(512) # Generate 512-bit RSA keys

def encrypt_message(message, public_key):
    return rsa.encrypt(message.encode('utf-8'), public_key)

def decrypt_message(encrypted_message, private_key):
    return rsa.decrypt(encrypted_message, private_key).decode('utf-8')

if __name__ == "__main__":
    # Generate RSA keys
    public_key, private_key = generate_keys()

    # Message to be encrypted
    message = input("Enter the message to be Encrypted: ")

    # Encrypt the message using the public key
    encrypted_message = encrypt_message(message, public_key)
    print(f"Encrypted Message: {encrypted_message}")

    # Decrypt the encrypted message using the private key
    decrypted_message = decrypt_message(encrypted_message, private_key)
    print(f"Decrypted Message: {decrypted_message}")
```

Output:

Enter the message to be Encrypted: Hello World!

EncryptedMessage:b'6\xef\xef\x1d\x942\xf3VV\x85\x8f\xef\xa4d\x05h\$\xcd\xcc\xa6\x10\xff\xde!|XX\xac\xa4\xa0\xa2\xf1\xda\xc\x92S\xc5q3\x11>\x95\xf5\xb4\xce3\xa9\x1c\x0e\x80\xc7\xde\x86\x88\xd2\xf4h\x81N\xa5K\x18\xd6'

Decrypted Message: Hello World!