Experiment No.: 10

Aim: - Implementation and analysis of RSA cryptosystem.

Software: - Virtual Labs

Theory: -

RSA Algorithm in Cryptography

RSA(**R**ivest-**S**hamir-**A**dleman) Algorithm is an **asymmetric** or **public-key cryptography** algorithm which means it works on two different keys: **Public Key** and **Private Key**. The Public Key is used for **encryption** and is known to everyone, while the Private Key is used for **decryption** and must be kept secret by the receiver. RSA Algorithm is named after Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman, who published the algorithm in 1977.

RSA Algorithm

RSA Algorithm is based on **factorization** of large number and **modular arithmetic** for encrypting and decrypting data. It consists of three main stages:

1. **Key Generation:** Creating Public and Private Keys
2. **Encryption:** Sender encrypts the data using Public Key to get **cipher text**.
3. **Decryption:** Decrypting the **cipher text** using Private Key to get the original data.

1. Key Generation

- Choose two large prime numbers, say **p** and **q**. These prime numbers should be kept secret.
- Calculate the product of primes, **n = p * q**. This product is part of the public as well as the private key.
- Calculate Euler Totient Function **Φ(n)** as **Φ(n) = Φ(p * q) = Φ(p) * Φ(q) = (p – 1) * (q – 1).**
- Choose encryption exponent **e**, such that
    - $1 < e < \Phi(n)$, and
- Calculate decryption exponent **d,** such that

- **(d * e) ≡ 1 mod Φ(n)**, that is d is modular multiplicative inverse of **e** mod Φ(n). Some common methods to calculate multiplicative inverse are: Extended Euclidean Algorithm, Fermat's Little Theorem, etc.
- We can have multiple values of d satisfying **(d * e) ≡ 1 mod Φ(n)** but it does not matter which value we choose as all of them are valid keys and will result into same message on decryption.

Finally, the **Public Key = (n, e)** and the **Private Key = (n, d)**.

2. Encryption

To encrypt a message **M**, it is first converted to numerical representation using ASCII and other encoding schemes. Now, use the public key (n, e) to encrypt the message and get the cipher text using the formula:

C = Me mod n, where C is the Cipher text and e and n are parts of public key.

3. Decryption

To decrypt the cipher text **C**, use the private key (n, d) and get the original data using the formula:

M = Cd mod n, where M is the message and d and n are parts of private key.

Example –

Key Generation –

Choose 2 prime numbers: p = 3, q = 11

Compute n=p×q=3×11=33n = p \times q = 3 \times 11 = 33n=p×q=3×11=33

Compute Euler's totient function: ϕ(n)=(p−1)×(q−1)=(3−1)×(11−1)=2×10=20

Choose e=7e = 7e=7 (must be coprime with 20)

Compute d (modular inverse of e mod ϕ(n)): d=7−1mod20=3(since 7×3≡1mod20)

Public Key: (e = 7, n = 33)
Private Key: (d = 3, n = 33)

Encryption –

Suppose we want to encrypt message M = 4:

$C=M^e \bmod n = 47 \bmod 33 = 16384 \bmod 33 = 31$

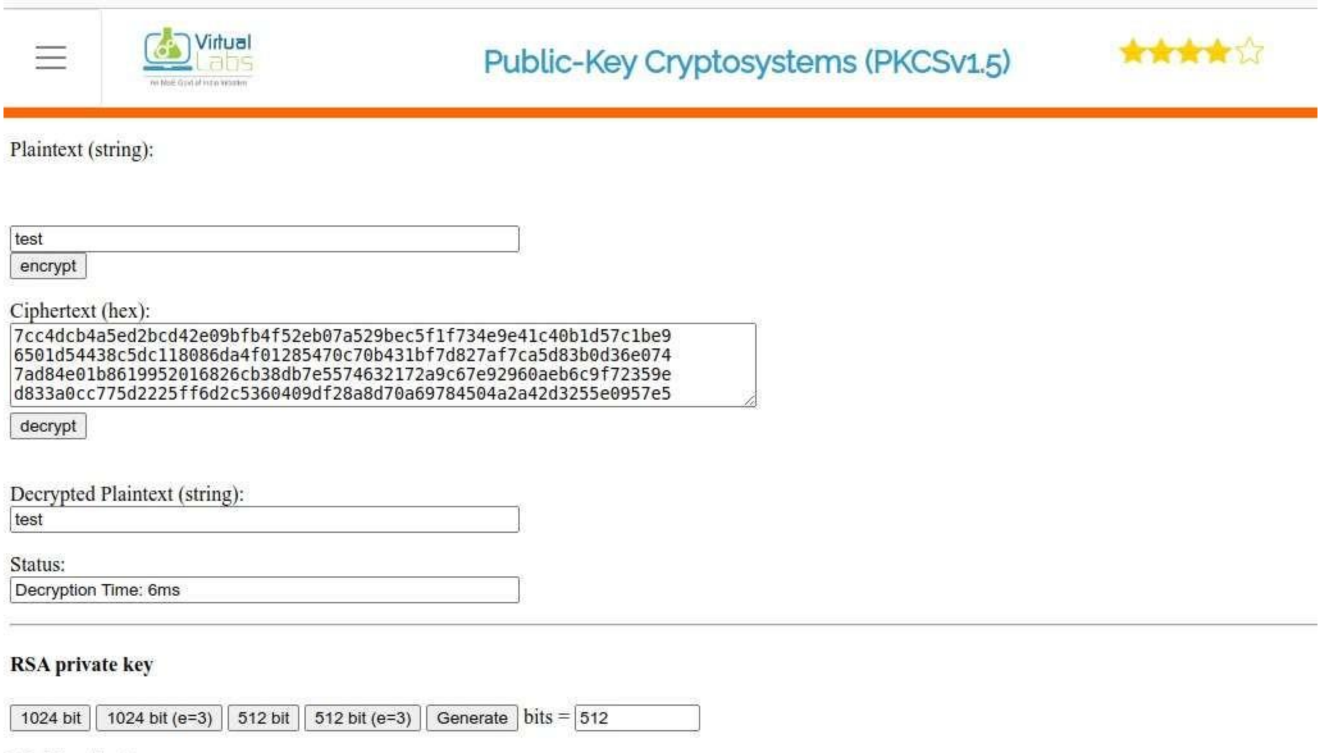
Encrypted message C = 31.

Decryption –

To decrypt C = 31, use the private key (d = 3, n = 33):

$M=C^d \bmod n=31^3 \bmod 33=29791 \bmod 33 = 4$

Decrypted message M = 4, which matches the original message.



Public-Key Cryptosystems (PKCSv1.5)

Modulus (hex):

```
a5261939975948bb7a58dffe5ff54e65f0498f9175f5a09288810b8975871e99
af3b5dd94057b0fc07535f5f97444504fa35169d461d0d30cf0192e307727c06
5168c788771c561a9400fb49175e9e6aa4e23fe11af69e9412dd23b0cb6684c4
c2429bce139e848ab26d0829073351f4acd36074eafd036a5eb83359d2a698d3
```

Public exponent (hex, F4=0x10001):

```
10001
```

Private exponent (hex):

```
8e9912f6d3645894e8d38cb58c0db81ff516cf4c7e5a14c7f1eddb1459d2cded
4d8d293fc97aee6aefb861859c8b6a3d1dfe710463e1f9ddc72048c09751971c
4a580aa51eb523357a3cc48d31cfad1d4a165066ed92d4748fb6571211da5cb1
4bc11b6e2df7c1a559e6d5ac1cd5c94703a22891464fba23d0d965086277a161
```

P (hex):

```
d090ce58a92c75233a6486cb0a9209bf3583b64f540c76f5294bb97d285eed33
aec220bde14b2417951178ac152ceab6da7090905b478195498b352048f15e7d
```

Q (hex):

```
cab575dc652bb66df15a0359609d51d1db184750c00c6698b90ef3465c996551
03edbf0d54c56aec0ce3c4d22592338092a126a0cc49f65a4a30d222b411e58f
```

D mod (P-1) (hex):

```
1a24bca8e273df2f0e47c199bbf678604e7df7215480c77c8db39f49b000ce2c
f7500038acfff5433b7d582a01f1826e6f4d42e1c57f5e1fef7b12aabc59fd25
```

D mod (Q-1) (hex):

```
3d06982efbbe47339e1f6d36b1216b8a741d410b0c662f54f7118b27b9a4ec9d
914337eb39841d8666f3034408cf94f5b62f11c402fc994fe15a05493150d9fd
```

1/Q mod P (hex):

```
3a3e731acd8960b7ff9eb81a7ff93bd1cfa74cbd56987db58b4594fb09c09084
db1734c8143f98b602b981aaa9243ca28deb69b5b280ee8dcee0fd2625e53250
```

Conclusion: - The RSA algorithm is a fundamental public-key cryptographic method widely used for encryption and digital signatures. Its security is based on the complexity of prime factorization, making it highly effective for protecting sensitive data.