



Experiment No. 3: Implementation and analysis of RSA cryptosystem

Aim: Implementation and analysis of RSA cryptosystem

Theory:

RSA was invented by Ron Rivest, Adi Shamir, and Len Adleman and hence, it is termed as RSA cryptosystem. We will see two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms.

Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below –

- Generate the RSA modulus (n)
- Select two large primes, p and q.
- Calculate $n=p \cdot q$. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.
- Find Derived Number (e)
 - Number e must be greater than 1 and less than $(p - 1)(q - 1)$.
 - There must be no common factor for e and $(p - 1)(q - 1)$ except for 1. In other words two numbers e and $(p - 1)(q - 1)$ are coprime.



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Academic Year 2023-24

- Form the public key
 - The pair of numbers (n, e) form the RSA public key and is made public.
 - Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.
- Generate the private key
 - Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
 - Number d is the inverse of e modulo $(p - 1)(q - 1)$. This means that d is the number less than $(p - 1)(q - 1)$ such that when multiplied by e, it is equal to 1 modulo $(p - 1)(q - 1)$.
 - This relationship is written mathematically as follows –

$$ed = 1 \pmod{(p - 1)(q - 1)}$$

The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output.



Example

An example of generating RSA Key pair is given below. (For ease of understanding, the primes p & q taken here are small values. Practically, these values are very high).

- Let two primes be $p = 7$ and $q = 13$. Thus, modulus $n = pq = 7 \times 13 = 91$.
- Select $e = 5$, which is a valid choice since there is no number that is common factor of 5 and $(p - 1)(q - 1) = 6 \times 12 = 72$, except for 1.
- The pair of numbers $(n, e) = (91, 5)$ forms the public key and can be made available to anyone whom we wish to be able to send us encrypted messages.
- Input $p = 7$, $q = 13$, and $e = 5$ to the Extended Euclidean Algorithm. The output will be $d = 29$.
- Check that the d calculated is correct by computing –

$$de = 29 \times 5 = 145 = 1 \bmod 72$$

- Hence, public key is $(91, 5)$ and private keys is $(91, 29)$.

Encryption and Decryption

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy.



RSA Encryption

- Suppose the sender wish to send some text message to someone whose public key is (n, e) .
- The sender then represents the plaintext as a series of numbers less than n .
- To encrypt the first plaintext P , which is a number modulo n . The encryption process is simple mathematical step as –

$$C = P^e \bmod n$$

- In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then reduced modulo n . This means that C is also a number less than n .
- Returning to our Key Generation example with plaintext $P = 10$, we get ciphertext C

$$C = 105 \bmod 91$$

RSA Decryption

- The decryption process for RSA is also very straightforward. Suppose that the receiver of public-key pair (n, e) has received a ciphertext C .



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Academic Year 2023-24

- Receiver raises C to the power of his private key d. The result modulo n will be the plaintext P.

$$\text{Plaintext} = C^d \bmod n$$

- Returning again to our numerical example, the ciphertext C = 82 would get decrypted to number 10 using private key 29 –

$$\text{Plaintext} = 82^{29} \bmod 91 = 10$$

Source Code:

```
import math

def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

def RSA(p: int, q: int, message: int):
    # calculating n
    n = p * q
    t = (p - 1) * (q - 1)

    for i in range(2, t):
        if gcd(i, t) == 1:
```



Vidyavardhini's College of Engineering & Technology
Department of Computer Engineering
Academic Year 2023-24

e = i

break

j = 0

while True:

 if (j * e) % t == 1:

 d = j

 break

 j += 1

ct = (message ** e) % n

print(f"Encrypted message is {ct}")

mes = (ct ** d) % n

print(f"Decrypted message is {mes}")

Testcase - 1

RSA(p=53, q=59, message=89)

Testcase - 2

RSA(p=3, q=7, message=12)



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Academic Year 2023-24

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kdbom\Desktop>python rsa.py
Encrypted message is 1394
Decrypted message is 89
Encrypted message is 3
Decrypted message is 12
```

Conclusion:

In conclusion, the RSA algorithm is implemented efficiently to encrypt and decrypt messages securely. By using a predefined value for the public exponent e (commonly 65537), the function ensures better security. This approach enhances the reliability of RSA encryption for secure communication and data protection.