MIT World Peace University

Information and Cyber Security

Assignment 4

Naman Soni Roll No. 10

Contents

1 Aim

Write a program using JAVA or Python or C++ to implement RSA asymmetric key algorithm.

2 Objective

To understand the concepts of public key and private key

3 Theory

3.1 *Number Theory*

- **EulerTotient Function:** Let n be a positive integer. The Euler totient function φ(n) is the number of positive integers less than or equal to n that are relatively prime to n. In other words, it is the number of integers k in the range $1 \leq k \leq n$ for which the greatest common divisor gcd(n,k) is equal to 1.

- **Euclidean Algorithm:** It is method for finding the GCD of two integers. It involves interactively finding the remainder when one number is divided by the other, and then replacing the larger number with the smaller number and the smaller number with the remainder. This process is repeated until the remainder is zero, at which point the GCD has been found. The Euclidean algorithm can be used to find the multiplicative inverse of a modulo n by finding the GCD of a and n and checking that it is 1. If it is, then the multiplicative inverse is given by the value of x in the last update step.

- **Extended Euclidean Algorithm:** The extended Euclidean algorithm is an extension to the Euclidean algorithm, and is used to find integers x and y such that ax+by= gcd(a,b). It is also used to find the multiplicative inverse of a modulo n, that is, the number x such that ax $\equiv$ 1 (mod n). The extended Euclidean algorithm is based on the observation that in the Euclidean algorithm, when a is replaced by a mod b, the new a and b are coprime, and the GCD does not change. This replacement can be repeated until the two numbers become equal, at which point the GCD has been found. The extended Euclidean algorithm goes one step further: at each step, the values of x and y are also updated. The updated values of x and y can be used to find the GCD of the original a and b by the equation ax+by= gcd(a,b). The extended Euclidean algorithm can be used to find the multiplicative inverse of a modulo n by finding the GCD of a and n and checking that it is 1. If it is, then the multiplicative inverse is given by the value of x in the last update step.

4 Language Used

Python

5 Code

```
import math


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


def isPrime(a):
for i in range(2, int(math.sqrt(a)) + 1):
if a % i == 0:
return False
return True


def extEuc(a, b):
old_r, r = a, b
old_s, s = 1, 0
old_t, t = 0, 1
while r != 0:
quotient = old_r // r
old_r, r = r, (old_r - (quotient * r))
old_s, s = s, (old_s - (quotient * s))
old_t, t = t, (old_t - (quotient * t))

return old_t


p, q = [int(i) for i in input("Enter Prime number p and q: ").split(" ")]

if isPrime(p) and isPrime(q):
n = p * q
totient_n = (p - 1) * (q - 1)
e = 2
while e < totient_n:
if gcd(totient_n, e) == 1:
break
e += 1
print("Value of e: ", e)
# calculating e inv
d = extEuc(totient_n, e) % totient_n
print("Value of d: ", d)
message = int(input("Enter message for encrypting"))
cypherText = pow(message, e) % n
print("Cypher text is ", cypherText)
messageDecrypt = pow(cypherText, d) % n
print("Message text is ", messageDecrypt)
else:
print("Entered number is not prime numbers")

```

Listing 1: Input

```
Enter Prime number p and q: 17 7
Value of e:  5
Value of d:  77
Enter message for encrypting 10
Cypher text is  40
Message text is  10
```

Listing 2: Output

6 Conclusion

Thus, learnt about the RSA encryption and decryption, and how to implement it using python.

7 FAQ's

1. Compare symmetric key cryptography and asymmetric key cryptography

   Ans.

   - In symmetric key cryptography, the same key is used for both encryption and decryption.
   - In asymmetric key cryptography, two different keys are used for encryption and decryption, which are Private key and Public Key respectively.

- symmetric key cryptography is faster.
- asymmetric key cryptography is slower than symmetric key cryptography.
- Symmetric key cryptography is used for bulk encryption.
- Asymmetric key cryptography is used for key exchange.
- In symmetric key cryptography, Requires managing a large no.of keys for each user.
- In asymmetric key cryptography, user has only one key pair.
- Symmetric key cryptography Example: DES, AES, 3DES.
- Asymmetric key cryptography Example: RSA, DSA, ECC.

2. Write advantages and disadvantages of RSA algorithm

   Ans. ***Advantages:***

   - **Security:** RSA utilizes two large prime numbers to create public and private keys for securing data transfers. The user's public key can be safely shared with anyone, while their private key remains secret. This makes it virtually impossible for an attacker to decrypt the message without the private key.
   - **Easy Implementation:** The RSA algorithm is relatively easy to implement compared to other encryp- tion algorithms. Since it relies on basic mathematical operations, it can be implemented using simple programming languages such as Python or Java.
   - **Flexible Key Management:** RSA supports a key length ranging from 1024 bits to 4096 bits. Longer key sizes provide better security but come with a computational cost. RSA algorithm also supports key rotation, which means users can change their keys regularly to enhance their protection.

   Disadvantages:

   - ***Slow:*** RSA is a relatively slow algorithm. It takes a long time to encrypt and decrypt data. This makes it unsuitable for applications that require high-speed data transfer.
   - ***Not Suitabel for Bulk Encryption:*** SA is not suitable for bulk encryption. It is best suited for encrypting small amounts of data.
   - ***Key Management:*** Key management is challenging in RSA encryption because large keys need higher processing power, which makes it difficult to generate, store and transmit safely.