Name: Devanshu Surana
Roll No.: 23
Prn:1032210755
Panel: C batch:C1

Lab A4: Implementation of RSA asymmetric key algorithm using python or java or C++

Objective of Lab

1. To introduce the fundamental concepts of discrete logarithmic problem, public key encryption , man in the middle attack and implement RSA Algorithm

Theory

1. Discrete Logarithm Problem:
The discrete logarithm problem is a fundamental concept in number theory and cryptography. It involves finding the integer 'x' in the equation 'a^x ≡ b (mod p),' where 'a' and 'p' are known values, and 'b' is the result of exponentiation modulo 'p.' Solving this problem is computationally difficult, especially for large prime numbers, which forms the basis of many cryptographic algorithms, including RSA and Diffie-Hellman.

2. Public Key Encryption:
Public key encryption is a cryptographic method that uses a pair of keys: a public key for encryption and a private key for decryption. The keys are mathematically related, but it is computationally infeasible to derive the private key from the public key. The security of public key encryption relies on the difficulty of certain mathematical problems, like the discrete logarithm problem.

3. Man-in-the-Middle Attack:
A man-in-the-middle attack occurs when an attacker intercepts communication between two parties and can eavesdrop on or alter the messages exchanged. The attacker typically poses as both parties to intercept, modify, or inject malicious data into the communication. Public key encryption and digital signatures are used to mitigate man-in-the-middle attacks.

Home Identity and access management
DEFINITION
RSA algorithm (Rivest-Shamir-Adleman)

Michael Cobb
By
Michael Cobb
What is the RSA algorithm (Rivest-Shamir-Adleman)?

The RSA algorithm (Rivest-Shamir-Adleman) is the basis of a cryptosystem -- a suite of cryptographic algorithms that are used for specific security services or purposes -- which enables public key encryption and is widely used to secure sensitive data, particularly when it is being sent over an insecure network such as the internet.

RSA was first publicly described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman of the Massachusetts Institute of Technology, though the 1973 creation of a public key algorithm by British mathematician Clifford Cocks was kept classified by the U.K.'s GCHQ until 1997.

Public key cryptography, also known as asymmetric cryptography, uses two different but mathematically linked keys -- one public and one private. The public key can be shared with everyone, whereas the private key must be kept secret.

Graphic displaying differences between symmetric vs. asymmetric encryption
RSA is a type of asymmetric encryption, which uses two different but linked keys.
In RSA cryptography, both the public and the private keys can encrypt a message. The opposite key from the one used to encrypt a message is used to decrypt it. This attribute is one reason why RSA has become the most widely used asymmetric algorithm: It provides a method to assure the confidentiality, integrity, authenticity, and non-repudiation of electronic communications and data storage.

Example of Using RSA Algorithm

- Choose p = 3 and q = 11
- Compute n = p * q = 3 * 11 = 33
- Compute φ(n) = (p - 1) * (q - 1) = 2 * 10 = 20
- Choose e such that 1 < e < φ(n) and e and φ (n) are coprime. Let e = 7
- Compute a value for d such that (d * e) % φ(n) = 1. One solution is d = 3 [(3 * 7) % 20 = 1]
- Public key is (e, n) => (7, 33)
- Private key is (d, n) => (3, 33)
- The encryption of $m = 2$ is $c = 2^7 \% 33 = 29$
- The decryption of $c = 29$ is $m = 29^3 \% 33 = 2$

Algorithm:

- Select two prime no's. Suppose P = 53 and Q = 59.
- Now First part of the Public key : n = P*Q = 3127.
- We also need a small exponent say e :
- But e Must be
- An integer.
- Not be a factor of Φ(n).
- 1 < e < Φ(n) [Φ(n) is discussed below],
- Let us now consider it to be equal to 3.
- Our Public Key is made of n and e

Code

```
import math
def gcd(a, h):
        temp = 0
        while(1):
                temp = a % h
                if (temp == 0):
                        return h
                a = h
                h = temp
p = 3
q = 7
n = p*q
e = 2
phi = (p-1)*(q-1)

while (e < phi):
        if(gcd(e, phi) == 1):
                break
        else:
                e = e+1
k = 2
d = (1 + (k*phi))/e

msg = 12.0

print("Message data = ", msg)

#  c = (msg ^ e) % n
c = pow(msg, e)
```

```
c = math.fmod(c, n)
print("Encrypted data = ", c)

# m = (c ^ d) % n
m = pow(c, d)
m = math.fmod(m, n)
print("Original Message Sent = ", m)
```

Output Screen shots

```
Message data =  12.0
Encrypted data =  3.0
Original Message Sent =  12.0
```

Conclusion: Thus, we have successfully learned and implemented RSA algorithm.

FAQs:

1. What is discrete logarithmic problem
2. What is man in middle attack
3. Explain RSA algorithm

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

Faculty: Engineering and Technology

School of Computer Engineering & Technology
Programme: B.Tech Computer Sc. & Engineering