**Experiment-10**

**Aim:** Use RSA for generation and verification of digital signature on file.

**Introduction:**

RSA digital signatures are like a digital seal of approval. They make sure that online stuff is real and has not been messed with. Here is how they work:

There are two keys, one for locking (public) and one for unlocking (private).

The private key makes a unique signature for the data, like a special stamp.

Other people can use the public key to check if the stamp is real, making sure nobody can deny sending it.

RSA works because it is hard to crack the code used for the stamp, and it is crucial for trust and safety online, like making sure documents are real and online payments are secure.

**Program (Source Code):**

// senders's authentication only

#include <bits/stdc++.h>

#include <cmath>

double PrivateP = 13;

double PrivateQ = 23;

double PrivateN = PrivateP \* PrivateQ;

double phi = (PrivateP - 1) \* (PrivateQ - 1);

double PrivateE = 2;

int T1 = 0;

int T2 = 1;

using namespace std;

double powerMod(int plaintext, double power, double n){

    if (int(power) == 0) {

        return 1;

    }

    double result = powerMod(plaintext, power / 2, n);

    result = int(result \* result) % int(n);

    if (int(power) % 2 == 1) {

        result = int(result \* plaintext) % int(n);

    }

    return result;

}

int gcd(int a, int b) {

    int temp;

    while (1){

        temp = a % b;

        if (temp == 0){

            return b;

        }

        a = b;

        b = temp;

    }

}

// to calculate inverse of a number

// to calculate value of d (inverse of e) modulo n

int extendedEuclidean(int a, int b){

    // a > b

    int A = a;

    int B = b;

    int Q = A/B;

    int R = A%B;

    int T = T1 - (T2\*Q);

    // cout<<Q<<" "<<A<<" "<<B<<" "<<R<<" "<<T1<<" "<<T2<<" "<<T<<" "<<endl;

    A = B;

    B = R;

    T1 = T2;

    T2 = T;

    if (B == 0){

        return T1;

    }

    else{

        return extendedEuclidean(A, B);

    }

}

int calculateE(int PrivateE){

    // calculating value of PrivateE

    while (PrivateE < phi){

        // e must be co-prime to phi and

        // smaller than phi.

        if (gcd(PrivateE, phi) == 1){

            break;

        }

        else{

            PrivateE++;

        }

    }

    return PrivateE;

}

// to generate MD1 and MD2 for message m

int hashingFunction(string m){

    int hash = 0;

    for (char c : m){

        hash = (hash \* 31) + c;

    }

    return hash;

}

// to generate digital signature for a md value generated in hashingFunction()

int generateDigitalSignature(int md){

    int k = 2;

    double PrivateD = extendedEuclidean(phi, PrivateE);

    while (PrivateD < 0){

        PrivateD += phi;

    }

    // cout<<phi<<endl;

    // cout<<PrivateE<<endl;

    // cout<<PrivateD<<endl;

    // cout<<PrivateN<<endl;

    // cout<<power<<endl;

    // double power = pow((int)md, (int)(PrivateD));

    // int power2 = int(power);

    // double digitalSignature = power2 % int(PrivateN);

    double digitalSignature = powerMod(md, PrivateD, PrivateN);

    return digitalSignature;

}

int decryptDigitalSignature(int digitalSignature){

    double PublicE = PrivateE;

    double PublicN = PrivateN;

    // double power = pow((int)digitalSignature, (int)(PublicE));

    // int power2 = int(power);

    // int md = power2 % (int)PublicN;

    double md0 = powerMod(int(digitalSignature), PublicE, PublicN);

    int md = int(md0);

    return md;

}

int checkAuthenticity(int originalMessage, int digitalSignature){

    int md = decryptDigitalSignature(digitalSignature);

    if (originalMessage == md){

        return true;

    }

    else{

        return false;

    }

}

int main() {

    // to calculate value of e which is needed by both Alice and Bob (sender & receiver)

    PrivateE = calculateE(PrivateE);

    int originalMessage = 130;

    cout<<"Original message: "<<originalMessage<<endl;

    int digitalSignature = generateDigitalSignature(originalMessage);

    cout<<"Digital Signature generated: "<<digitalSignature<<endl;

    if (checkAuthenticity(originalMessage, digitalSignature)){

        cout<<"Authenication Successful!"<<endl;

    }

    else{

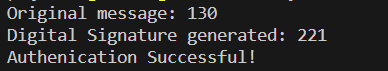
        cout<<"Authentication Failed!"<<endl;

    }

    return 0;

}

**Output (Program):**

****

**Cryptanalysis:**

**Advantages of RSA Technique:**

1. Security – RSA algorithm is a very secure method for encrypting and decrypting sensitive information. It uses the properties of large prime numbers to make it difficult for outsiders to crack the code.
2. Digital Signatures – RSA algorithm can be used to create digital signatures, which can help to verify the authenticity of digital documents.
3. Speed – RSA algorithm is relatively fast and efficient, making it suitable for use in real-time applications.

**Disadvantages of RSA Technique:**

1. Key Size – RSA algorithm requires large prime numbers as part of the encryption process. The larger the prime numbers, the more secure the encryption, but it also increases the key size and processing time.
2. Vulnerability to Quantum Computing – RSA algorithm is vulnerable to attacks by quantum computers, which can potentially break the encryption.

**Applications:**

**1) Secure Communication:** Digital signatures are used to ensure that messages or data exchanged between parties have not been tampered with during transmission. This is essential for secure email communication, instant messaging, and online chats.

**2) Software and Firmware Authentication:** Digital signatures are employed to verify the authenticity and integrity of software applications, updates, and firmware. This prevents the installation of malicious or tampered software.

**3) Document Authentication:** In legal and business contexts, digital signatures are used to verify the authenticity of electronic documents, contracts, and agreements. This eliminates the need for physical signatures and provides a secure method for electronic document management.

**4) E-commerce and Online Transactions:** Digital signatures play a crucial role in online payment systems and e-commerce. They ensure that online transactions are secure and that the transaction details are unaltered.

**References:**

1. GeeksforGeeks
2. www.javatpoint.com
3. www.cryptool.org/en/cto/hill