NAME: Sarvesh Patil
CLASS: D15A
ROLL NO: 52

LAB 06

LAB 06: Write a program to implement and analyze RSA cryptosystem

ROLL NO	52
NAME	Sarvesh Patil
CLASS	D15A
SUBJECT	Internet Security Lab
LO MAPPED	LO1: To apply the knowledge of symmetric cryptography to implement classical ciphers. LO2: Demonstrate Key management, distribution, and user authentication.

NAME: Sarvesh Patil
CLASS: D15A
ROLL NO: 52

AIM:
Write a program to implement and analyze the RSA cryptosystem

INTRODUCTION:

RSA:
RSA, or in other words Rivest–Shamir–Adleman, is an asymmetric cryptographic algorithm. It differs from symmetric algorithms like DES or AES by having two keys. A public key that we can share with anyone is used to encrypt data. And a private one that we keep only for ourselves and it's used for decrypting the data.

RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes, the Public Key is given to everyone and the Private key is kept private.

An example of asymmetric cryptography:
1. A client (for example browser) sends its public key to the server and requests for some data.
2. The server encrypts the data using the client's public key and sends the encrypted data.
3. Client receives this data and decrypts it.

Since this is asymmetric, nobody else except the browser can decrypt the data even if a third party has the public key of the browser.

The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is the multiplication of two large prime numbers. And private keys are also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies in the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken in the near future. But till now it seems to be an infeasible task.

RSA algorithm is popular exponentiation in a finite field over integers including prime numbers. The integers used by this method are sufficiently large making it difficult to solve. There are two sets of keys in this algorithm: a private key and a public key.

RSA is an encryption algorithm, used to securely transmit messages over the internet. It is based on the principle that it is easy to multiply large numbers, but factoring in large numbers is very difficult. For example, it is easy to check that 31 and 37 multiply by 1147, but trying to find the factors of 1147 is a much longer process.

A prime number is one that is divisible only by one and itself. For example, 3 is a prime number, because it can be divided only by 1 or 3. But 4 is not a prime number, because other than by 1 and 4, it can also be divided by 2. Likewise, 5, 7, 11, 13, 17….are prime numbers whereas 6, 8, 9, 10, 12 are non-prime numbers.

The RSA Algorithm depends on the mathematical part that it is simple to discover and multiply large prime numbers together, but it is intensely complex to factor their product. RSA supports both confidentiality (encryption with a public key and decrypting with a private key) and digital signing uniformly protected.

RSA Information Security pioneered and marketed the technology that creates it possible to connect and transfer data and documents securely on the web and creates and authenticate the identity of virtual trading partners—developments important to the widespread acceptance of digital commerce.

RSA algorithm is a public key encryption technology and is considered the most secure way of encryption. It was invented by Rivest, Shamir, and Adleman in the year 1978 and hence the name the RSA algorithm.

The RSA algorithm holds the following features –

- RSA algorithm is popular exponentiation in a finite field over integers including prime numbers.
- The integers used by this method are sufficiently large making it difficult to solve.
- There are two sets of keys in this algorithm: a private key and public key.

You will have to go through the following steps to work on the RSA algorithm –

Step 1: Generate the RSA modulus
The initial procedure begins with the selection of two prime numbers namely p and q, and then calculating their product N, as shown –

$N=p*q$
Here, let N be the specified large number.

Step 2: Derived Number (e)
Consider number e as a derived number that should be greater than 1 and less than (p-1) and (q-1). The primary condition will be that there should be no common factor of (p-1) and (q-1) except 1

Step 3: Public key
The specified pair of numbers n and e forms the RSA public key and it is made public.

Step 4: Private Key
Private Key d is calculated from the numbers p, q, and e. The mathematical relationship between the numbers is as follows –

$ed = 1 \bmod (p-1)(q-1)$
The above formula is the basic formula for the Extended Euclidean Algorithm, which takes p and q as the input parameters.

NAME: Sarvesh Patil
CLASS: D15A
ROLL NO: 52

Encryption Formula

Consider a sender who sends the plain text message to someone whose public key is (n,e). To encrypt the plain text message in the given scenario, use the following syntax –

C = Pe mod n

Decryption Formula

The decryption process is very straightforward and includes analytics for calculation in a systematic approach. Considering receiver C has the private key d, the resulting modulus will be calculated as –

Plaintext = Cd mod n

RSA Encryption Algorithm

RSA encryption algorithm is a type of public-key encryption algorithm. To better understand RSA, let's first understand what is a public-key encryption algorithm.

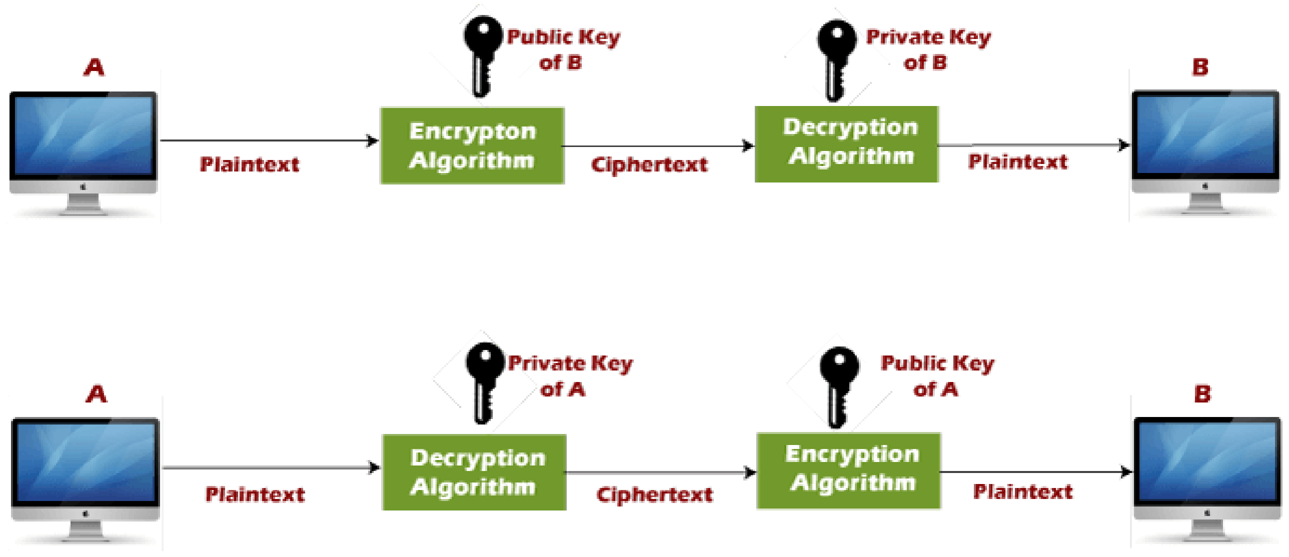
Public key encryption algorithm:

Public Key encryption algorithm is also called the Asymmetric algorithm. Asymmetric algorithms are those algorithms in which the sender and receiver use different keys for encryption and decryption. Each sender is assigned a pair of keys:

- Public key
- Private key

The Public key is used for encryption, and the Private Key is used for decryption. Decryption cannot be done using a public key. The two keys are linked, but the private key cannot be derived from the public key. The public key is well known, but the private key is secret and it is known only to the user who owns the key. It means that everybody can send a message to the user using the user's public key. But only the user can decrypt the message using his private key.

The Public key algorithm operates in the following manner:



Encryption/decryption using public/private keys

NAME: Sarvesh Patil
CLASS: D15A
ROLL NO: 52

- The data to be sent is encrypted by sender A using the public key of the intended receiver
- B decrypts the received ciphertext using its private key, which is known only to B. B replies to A encrypting its message using A's public key.
- A decrypts the received ciphertext using its private key, which is known only to him.

RSA algorithm uses the following procedure to generate public and private keys:

1. Select two large prime numbers, p, and q.
2. Multiply these numbers to find n = p x q, where n is called the modulus for encryption and decryption.
3. Choose a number e less than n, such that n is relatively prime to (p - 1) x (q -1). It means that e and (p - 1) x (q - 1) have no common factor except 1. Choose "e" such that 1<e < φ (n), e is prime to φ (n),
   gcd (e,d(n)) =1
4. If n = p x q, then the public key is <e, n>. A plaintext message m is encrypted using the public key <e, n>. To find ciphertext from the plain text following formula is used to get ciphertext C.
   C = me mod n
   Here, m must be less than n. A larger message (>n) is treated as a concatenation of messages, each of which is encrypted separately.
5. To determine the private key, we use the following formula to calculate the d such that:
   De mod {(p - 1) x (q - 1)} = 1
   Or
   De mod φ (n) = 1
6. The private key is <d, n>. A ciphertext message c is decrypted using private key <d, n>. To calculate plain text m from the ciphertext c following formula is used to get plain text m.
   m = cd mod n

ALGORITHM:

Step 1: Choose two prime numbers p and q.
Step 2: Calculate n = p*q
Step 3: Calculate ϕ(n) = (p – 1) * (q – 1)
Step 4: Choose e such that gcd(e , ϕ(n)) = 1
Step 5: Calculate d such that e*d mod ϕ(n) = 1
Step 6: Public Key {e,n} Private Key {d,n}
Step 7: Cipher text C = Pe mod n where P = plaintext
Step 8: For Decryption D = Dd mod n where D will give back the plaintext.

NAME: Sarvesh Patil
CLASS: D15A
ROLL NO: 52

PROGRAM:

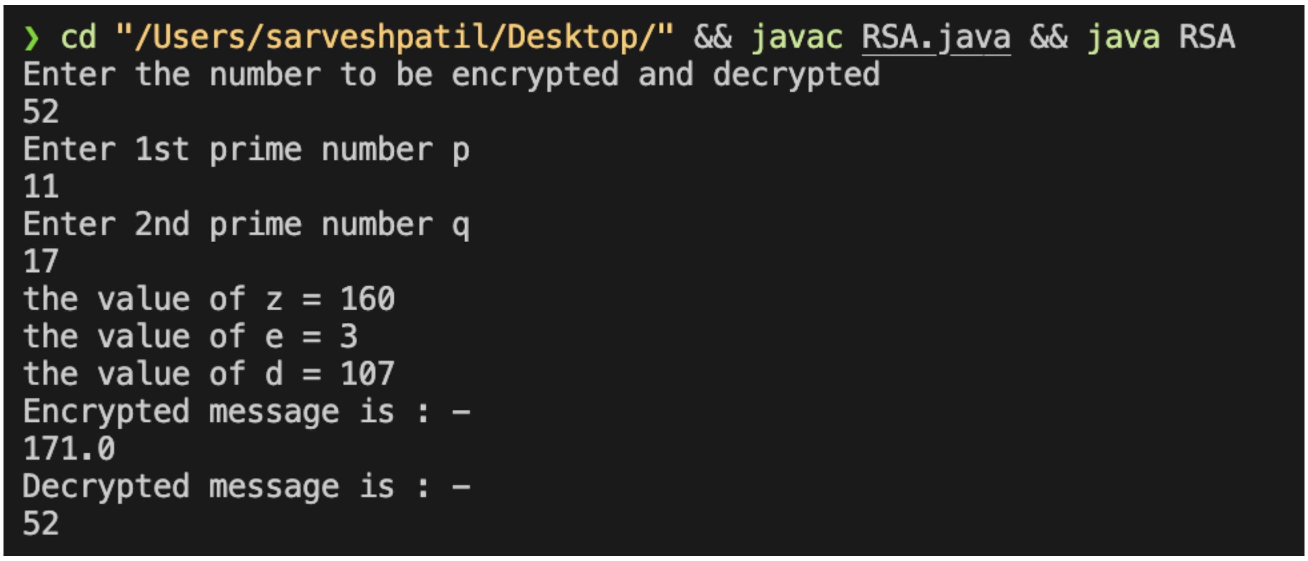
```
import java.util.*;
import java.math.*;

class RSA
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		int p,q,n,z,d=0,e,i;
		System.out.println("Enter the number to be encrypted and decrypted");
		int msg=sc.nextInt();
		double c;
		BigInteger msgback;
		System.out.println("Enter 1st prime number p");
		p=sc.nextInt();
		System.out.println("Enter 2nd prime number q");
		q=sc.nextInt();

		n=p*q;
		z=(p-1)*(q-1);
		System.out.println("the value of z = "+z);

		for(e=2;e<z;e++)
		{
			if(gcd(e,z)==1)          // e is for public key exponent
			{
				break;
			}
		}
		System.out.println("the value of e = "+e);
		for(i=0;i<=9;i++)
		{
			int x=1+(i*z);
			if(x%e==0)      //d is for private key exponent
			{
				d=x/e;
				break;
			}
		}
		System.out.println("the value of d = "+d);
		c=(Math.pow(msg,e))%n;
		System.out.println("Encrypted message is : -");
		System.out.println(c);
	  //converting int value of n to BigInteger
```

```
        BigInteger N = BigInteger.valueOf(n);
        //converting float value of c to BigInteger
        BigInteger C = BigDecimal.valueOf(c).toBigInteger();
        msgback = (C.pow(d)).mod(N);
        System.out.println("Decrypted message is : -");
        System.out.println(msgback);

    }

    static int gcd(int e, int z)
    {
        if(e==0)
            return z;
        else
            return gcd(z%e,e);
    }
}
```

NAME: Sarvesh Patil
CLASS: D15A
ROLL NO: 52

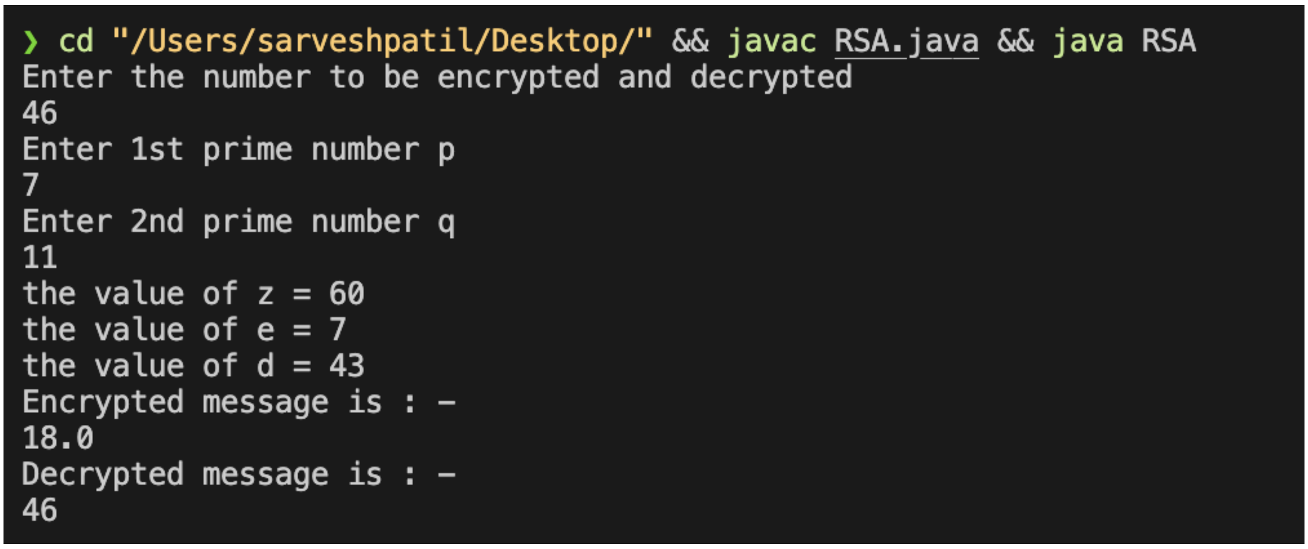
RESULTS:

TEST CASE 1:
Plaintext: 52
Prime numbers p and q: 11, 17



```
❯ cd "/Users/sarveshpatil/Desktop/" && javac RSA.java && java RSA
Enter the number to be encrypted and decrypted
52
Enter 1st prime number p
11
Enter 2nd prime number q
17
the value of z = 160
the value of e = 3
the value of d = 107
Encrypted message is : -
171.0
Decrypted message is : -
52
```

TEST CASE 2:
Plaintext: 46
Prime numbers p and q: 7, 11



```
❯ cd "/Users/sarveshpatil/Desktop/" && javac RSA.java && java RSA
Enter the number to be encrypted and decrypted
46
Enter 1st prime number p
7
Enter 2nd prime number q
11
the value of z = 60
the value of e = 7
the value of d = 43
Encrypted message is : -
18.0
Decrypted message is : -
46
```

CONCLUSION:
We have successfully studied and implemented the RSA algorithm.