Practical No – 02

RSA Encryption and Decryption

Aim: Implement the RSA algorithm for public-key encryption and decryption, and explore its properties and security considerations.

Source Code :

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

public class RSA { public static void

main(String args[]) {

    int p, q, n, z, d = 0, e, i;

    double c;

    BigInteger msgback;

    p = 5; q = 11; int

    msg = 12; n = p *

    q; z = (p - 1) * (q -

    1);

    System.out.println("The value of z = " + z);

    for(e = 2; e < z; e++){ //e is public key

    exponent if(gcd(e, z) == 1){ 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 = x / e;

        break;
}

    }

    System.out.println("The value of d = " + d); c

    = (Math.pow(msg, e)) % n;
```

```
        System.out.println("Encrypted message is : " + c);
        BigInteger N = BigInteger.valueOf(n); BigInteger C =
        BigDecimal.valueOf(c).toBigInteger(); msgback =
        (C.pow(d)).mod(N);
        System.out.println("Decryption message is : " + msgback);
    }
    static int gcd(int e, int z){
        if(e == 0){ return z;
        }
        else{ return gcd(z % e,
            e);
        }
    }

}
```

Output :