Asymmetric Encryption using Cryptographic Algorithms and RSA algorithms

Review Presentation – 1

By

Team LAANS

Under the Guidance of

Dr. Vishnu Srinivasa Murthy Y

School of Computer Science and Engineering

Vellore Institute of Technology (VIT),

Vellore, Tamil Nadu – 632 014.

Team details

Sl. No.	Regd. No.	Student Name	Role
1	19BCB0031	Lakshit Manish Sanghrajka	Coding and Research
2	19BCE0883	Aviral Goyal	Coding and Research
3	19BCE2362	Abhinav Dholi	Coding and Research
4	19BCE2543	Nehul jindal	Coding and Research
5	19BCB0083	Saksham Patel	Coding and Research

Introduction

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key.

This has two important consequences:

1. Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.

2. A message can be “signed” using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in “electronic mail” and “electronic funds transfer” systems.

Motivation

In today’s digital era , the significance of security is at its peak and daily hundreds of security gets breached and further more higher level of security gets implemented.

Right from banking transaction , seat reservation , transportation services like Ola and Uber , online shopping we don’t use physical money, we use the digital mode of payment i.e. cashless method which requires highly secured sites and payment gateways.

Also this is an information Age where information is equally valuable as money and we need to store our data securely.

Also the most used method OTP which is considered as a common security practice can be acquired and hacked.

So to prevent data theft and all other information which data stored is capable of , data encryption and decryption plays a pivotal role in today’s world to make passwords more strong and unbreakable.

Applications

1) **ATM’S and Banking systems**
2) **Web transactions**
3) **Cloud Services**

Proposed Methodology

The RSA algorithm involves four steps:

1.Key generation 2. Key distribution 3. Encryption 4. Decryption.

A basic principle behind RSA is the observation that it is practical to find three very large positive integers e, d and n such that with modular exponentiation for all integer m (with $0 \le m < n$): and that even knowing e and n or even m it can be extremely difficult to find d. In addition, for some operations it is convenient that the order of the two exponentiations can be changed and that this relation also implies:

RSA involves a public key and a private key.

The public key can be known by everyone, and it is used for encrypting messages. The intention is that messages encrypted with the public key can only be decrypted in a reasonable amount of time by using the private key.

The public key is represented by the integers n and e; and, the private key, by the integer d (although n is also used during the decryption process.

Thus, it might be considered to be a part of the private key, too). m represents the message (previously prepared with a certain technique explained below).

EXAMPLE

Generating Public Key:

Select two prime no's -> Suppose P = 53 and Q = 59.

Now First part of the Public key:n = P*Q = 3127.

$\Phi(n) = (P-1)\ (Q-1)$ so, $\Phi(n) = 3016$

e: $1 < e < \Phi(n)$

Let us now consider it to be equal to 3.

Our Public Key is made of n and e

Generating Private Key:

Now calculate Private Key,

d: $d = (k*\Phi(n) + 1) / e$ for some integer k

For k = 2, value of d is 2011.

Now we are ready with our – Public Key (n = 3127 and e = 3) and Private Key (d = 2011)

Encryption and Decryption:

Now we will encrypt “HI”:

Convert letters to numbers: H = 8 and I = 9

Thus, Encrypted Data c = 89e mod n. Thus, our Encrypted Data comes out to be 1394

Now we will decrypt 1349: Decrypted Data = cd mod n.

Thus, our Encrypted Data comes out to be 89 -> 8 = H and I = 9 i.e. "HI".

Any Novelty in your work?

This project proposes a novel way for reducing the impact of two constraints crucial to cryptographic algorithms i.e. time and computing power.

Tools used and needed

- CODEBLOCKS

Code of RSA Algorithm

```
#include<iostream>

#include<cstdio>

#include<math.h>

#include<cstring>

#include<stdlib.h>

#include<fstream>

#include<cstdio>

#include<time.h>

using namespace std;
```

```
long int p, q, n, t, flag, e[100], d[100], temp[100], j, m[100],
en[100],i,enckey,deckey;

 char msg[100];

 int prime(long int);

 void ce();

 long int cd(long int);

 void encrypt();

 void decrypt();

 int prime(long int pr)
```

```
{
  int i;
  j = sqrt(pr);
  for (i = 2; i <= j; i++)
  {
        if (pr % i == 0)  return 0;
  }
  return 1;
}
```

```
int primegenerator(int low,int high)
{
  int primearr[1000],i,j=0,flag=0,sizert,secret;  srand(time(0));
  while (low < high)
  {
     flag = 0;
     for(i = 2; i <= low/2; ++i)
     {
        if(low % i == 0)
        {
           flag = 1;  break;
        }
     }
     if (flag == 0)
        {
           primearr[j]=low;
           ++j;
        }
        ++low;
  }
  secret = rand() % (j);
  return(primearr[secret]);
}
```

```
int main()
{
  int choice;
  cout<<"\nLevel of Encryption";
  cout<<"\n1.Mobile/Portable";
  cout<<"\n2.Intermediate";
  cout<<"\n3.Crucial\n";
  cout<<"\nEnter your choice:";
  cin>>choice;
```

```
switch(choice)
{
   case 1:
   p=primegenerator(63,126);
   q=primegenerator(65,152);
   break;
   case 2:
   p=primegenerator(75,200);
   q=primegenerator(73,205);
   break;
   case 3:
   p=primegenerator(124,263);
   q=primegenerator(139,230);
   break;
   default:
   cout<<"Wrong Choice";
   exit(1);
}
//cin>>p>>q;
cout << "\nENTER MESSAGE\n";
fflush(stdin);
gets(msg);
for (i = 0; msg[i] != '\0'; i++)
```

```
m[i] = msg[i];
n = p * q;
t = (p - 1) * (q - 1);
ce();

fstream f("public.dat",ios::out | ios::binary);  fstream
f1("private.dat",ios::out | ios::binary);

srand(time(0));

int ee= rand()%20;

enckey= e[ee];

deckey= d[ee];
```

```
  f<<enckey<<n;

  f1<<deckey<<n;

  f.close();

  f1.close();
  cout << "\nPOSSIBLE VALUES OF e AND d ARE\n";

  for (i = 0; i < j - 1; i++)

              cout << e[i] << "\t" << d[i] << "\n";

encrypt();

decrypt();

return 0;

getchar();
```

```
void ce()
{
  int k;    k = 0;
  for (i = 2; i < t; i++)
  {
      if (t % i == 0)
      continue;
    flag = prime(i);
    if (flag == 1 && i != p && i != q)
    {
      e[k] = i;
      flag = cd(e[k]);
```

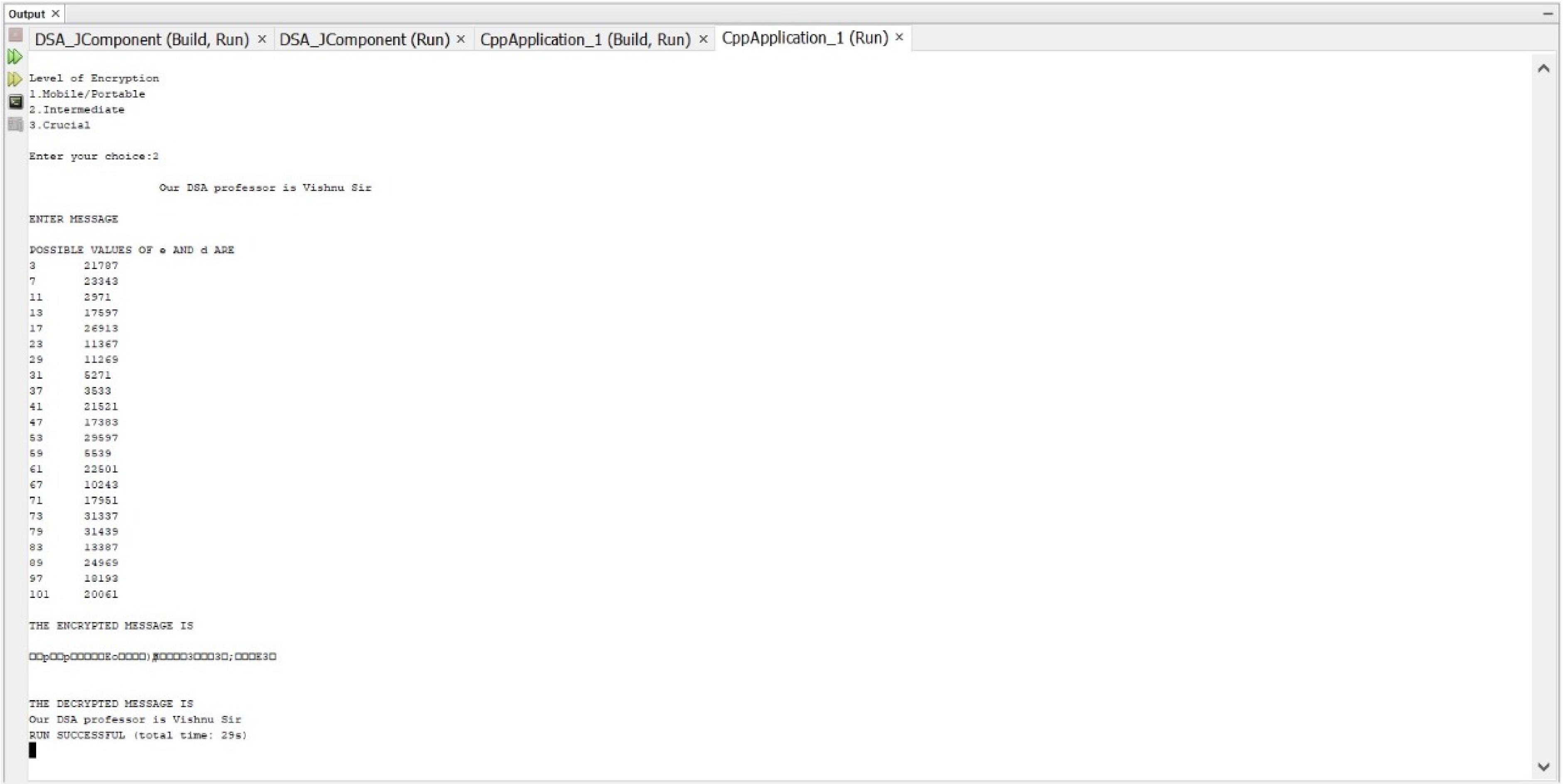
```
        if (flag > 0)
        {
           d[k] = flag;  k++;
        }
        if (k == 99)
           Break;
           }
  }
}
long int cd(long int x)
{
  long int k = 1;
  while (1)
  {
     k = k + t;
        if (k % x == 0)  return (k / x);
  }
}
void encrypt()
{
  ofstream f;
  f.open("enc.dat",ios::out | ios::binary);
  long int pt, ct, key = e[4], k, len;
  i = 0;
  len = strlen(msg);
  while (i != len)
```

```
{
   pt = m[i];
   pt = pt - 96;   k = 1;
   for (j = 0; j < key; j++)
   {
      k = k * pt;   k = k % n;
   }
   temp[i] = k;  ct = k + 96;  en[i] = ct;
   i++;
}
en[i] = -1;
char writestr[500];
```

```
  for(i=0 ; en[i] != -1 ; i++)
  {
    writestr[i]=en[i];
  }
  f.write(writestr,strlen(writestr));  f.close();
  cout << "\nTHE ENCRYPTED MESSAGE IS\n";
  cout<<"\n"<<writestr<<"\n";
}
void decrypt()
{
```

```
long int pt, ct, key = d[4], k;
i = 0;
while (en[i] != -1)
{
      ct = temp[i];   k = 1;
  for (j = 0; j < key; j++)
  {
     k = k * ct;   k = k % n;
  }
  pt = k + 96;
  m[i] = pt;
  i++;
  }
  m[i] = -1;
  cout << "\n\nTHE DECRYPTED MESSAGE IS\n";
  for (i = 0; m[i] != -1; i++)
    printf("%c", m[i]);
}
```

Output of program



```
Level of Encryption
1.Mobile/Portable
2.Intermediate
3.Crucial

Enter your choice:2

                   Our DSA professor is Vishnu Sir

ENTER MESSAGE

POSSIBLE VALUES OF e AND d ARE
3       21787
7       23343
11      2971
13      17597
17      26913
23      11367
29      11269
31      5271
37      3533
41      21521
47      17383
53      29597
59      5539
61      22501
67      10243
71      17951
73      31337
79      31439
83      13387
89      24969
97      18193
101     20061

THE ENCRYPTED MESSAGE IS

□□p□□p□□□□□Eo□□□□)□□□□□3□□□3□;□□□E3□


THE DECRYPTED MESSAGE IS
Our DSA professor is Vishnu Sir
RUN SUCCESSFUL (total time: 29s)
```

Conclusion

Even though RSA is the most used cryptography algorithm today, it has certain limitations which need to be taken into consideration for RSA to continue to be the best and research has to be done into making RSA quantum resistant.

There is a need now more than ever for studies to be conducted in the area of quantum encryption methods resistant to quantum computers as it will soon replace the current encryption systems. Development of qCrypt isn’t enough, but it’s a start. However, we need more research into quantum resistant encryption systems.

References

1. **R.L. Rivest, A. Shamir, and L. Adleman**

A Method for Obtaining Digital Signatures and Public-Key Cryptosystems (1977, 2000)

https://people.csail.mit.edu/rivest/Rsapaper.pdf

2. **Shireen Nisha, Mohammed Farik**

RSA Public Key Cryptography Algorithm – A Review (2017)

http://www.ijstr.org/final-print/july2017/Rsa-Public-Key-Cryptography-Algorithm-A- Review.pdf

3.Evgeny Milanov

The RSA Algorithm (2009)

https://sites.math.washington.edu/~morrow/336_09/papers/Yevgeny.pdf

4.RSA (cryptosystem)

https://en.wikipedia.org/wiki/RSA_(cryptosystem)

REVIEW 3 LINK

https://drive.google.com/file/d/1l-cqLTLF3Awzv-MEO7WqPSPaAFBU1k3J/view?usp=sharing

Thank You