IMPLEMENTING RSA ALGORITHM TO SECURE DATA FLOW IN THE COMPUTER NETWORKS

[1]Mr. Desta Dana, [2]Dr. T.MuraliKirishina
[1]Lecturer, [2]Assist. Prof. of Information Technology, Wolaita Sodo University, Ethiopia
[1]destaju@yahoo.com, [2]murali2007@gmail.com

Abstract:**The data flows between two machines that i.e. source and destination the data should be locked in the two keys that are called Private and public keys. The algorithm that used to make strong security in data flow inside the computer network; in this research work is RSA. The algorithmused to encrypt and decrypt the flow of data between the source (A) and Destination(B). If the source (A) sends data to Destination (B) then my experiment could automates first A encrypts the message or Plaintext by using sources (A) private key and Public keys of A.In general, RSA data encrypting method useful to secure and lock data in the computer network.**

Keywords: **Computer Network, Cipher Text, Data Flow, Decrypt, Encrypt, Implementing, Plaintext, Public key, Private Key, RSA Algorithm, Security**

1. INTRODUCTION

Data is the most expensive recourses for any company in the world. Assume some body send his money from one account to another but the transaction between the two bank industries is not a money but its data. Not only that but also an organizations have ongoing and planned projects documents so which is also data. Data is the most valuable materials and resources for the any organizations if its organized, documented and used in safe environments.Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites. Data security also protects data from data or file corruption in the computer networks.

Data security is an essential aspect of IT for organizations of every size and type.[2,3]. Network Security is an organization's strategy and provisions for ensuring the security of its assets and of all network traffic. Network security is manifested in an implementation of security hardware, and software to protect enhance the security of data inside the computer Securing the flow of data inside and outside the organizations requires some of the techniques to make it as hidden[6,5]. One of the most known algorithms we implemented in our research work is RSA algorithm. The RSA Algorithm is one of data encrypting mechanisms which uses Encryptions and decryptions are carried out using two different keys. The two keys in such a key pair are referred to as the public key and the private key. [4,5]

2. METHODOLOGY

We implemented in our study how can the RSA works and how it can be secured and hides the data in the network. RSA is the most known data encrypting methods and which uses the two keys called public and private keys.The algorithm implemented based on the diagram as it mentioned in below [1]

Assume the communications between two computers in the network which are called computer A and Computer B: Computer A, if wanting to communicate confidentially with Computer. B, can encrypt a message using B's publicly available key. Such a communication would only be decipherable by B as only B would have access to the corresponding private key.

This is illustrated by the top communication link in **Fig.1.**Computer A, if wanting to send an authenticated message toComputer B, would encrypt the message with A's own private key. Since this message would only be decipherable with A's public key, that would establish the authenticity of the message meaning that A was indeed the source of the message.

This we illustrated by the middle communication link in **Fig.2**. The communication link at the bottom of Fig.3 shows how public-key encryption can be used to provide both confidentiality and authentication at the same time. In Fig1, A's public and private keys are designated PUA andPRA. B's public and private keys are designated PUB and PRB. As shown at the bottom of **Fig.3**, let's say that A wants to send a message M to B with both authentication and confidentiality [1, 3].

In general, Authentication and authorization go hand in hand. Users must be authenticated before carrying out the activity they are authorized to perform. Security is strong when the means of authentication cannot later be refuted the user cannot later deny that he or she performed the activity.

Encrypting the message by using public and private keys: [When only confidentiality is needed:]

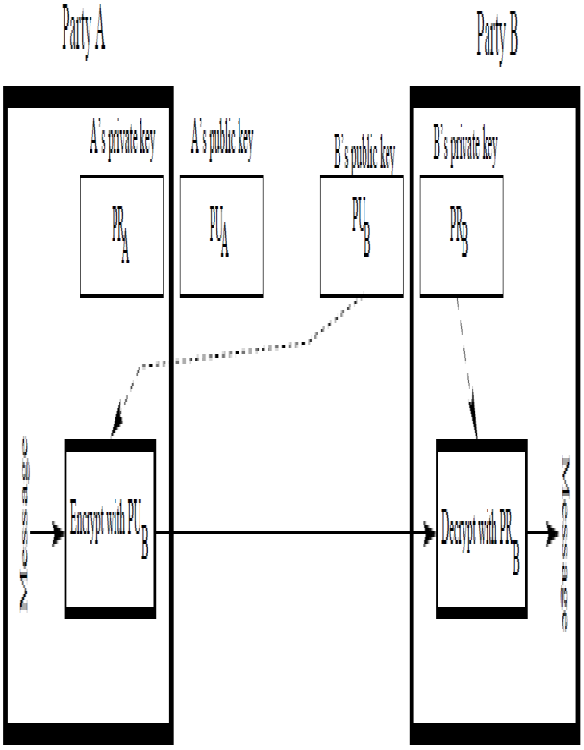


Fig1: Encrypting Plaintext (Message) [1]

Encrypting the message by using public and private keys :[When only authentication is needed:]

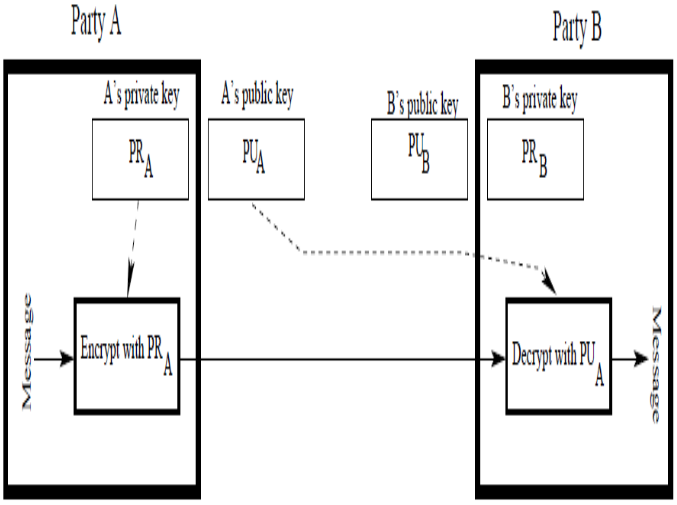


Fig2: Decrypting Cipher text [1]

Key exchanges between the Source and Destination [Both confidentiality and authentication are needed:]

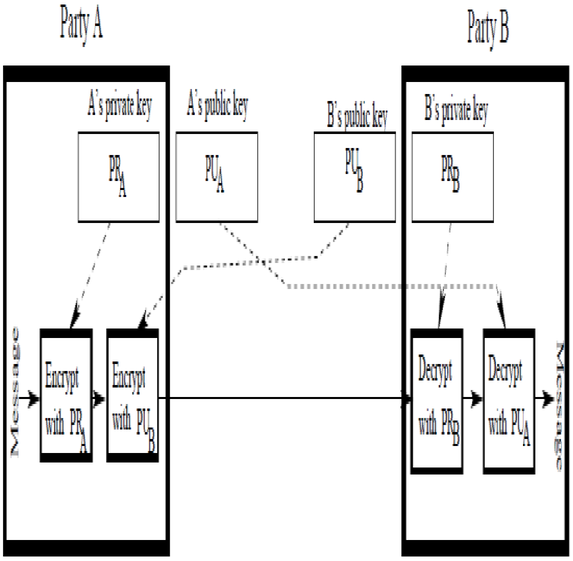


Fig3: Public and Private keys exchanges between the source and destination [1]

From the above Ciphering algorithm Diagrams

In both sender and receiver must know the value of n. The sender knows the value of e, and only receiver knows the value of d. Thus, this is a public-key encryption algorithm with a public key of PU={e, n}, and a private key of PR={d,n}. For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:

A. PROOFING THE RSA ALGORITHM.

$$C = M^e \quad mod \quad n$$

(1)

$$M = C^d \quad mod \quad n = (M^e)^d \quad mod \quad n = M^{(ed)} \quad mod \quad n$$

(2)

Meaning: M= Message (Plaintext) called original messages

C= Cipher Text (Encrypted Message)

n= is the product of two prime numbers.

B. DESCRIPTION OF THE ALGORITHM

1. It is possible to find values of e,d,n such that $M^{ed} = M \bmod n$ for all M<n
2. It is relatively easy to calculate M^e and C^d for all values of M<n
3. It is infeasible to determine d given e and d.

For now, we focus on the 1st requirement and consider the other questions later. We need to find a relationship of the form

$$M^{ed} = M \bmod n$$

A corollary to Euler's theorem

(For every a and n that are relatively prime

$$a^{\varphi(n)} \equiv 1 \bmod n$$

(3)

where $\varphi(n)$ is the Euler's totient function number of positive integers less than n and relatively prime to n), fits the bill:

Given two prime numbers, p and q, and two integers, n and m, such that n=p* q and 0<m<n, and arbitrary integer k, the following relationship holds:

$$m^{k\varphi(n)+1} = m^{k(p-1)(q-1)+1} \equiv m \bmod n$$

(4)

(As far as for p,q prime, $\varphi(n) = (p-1)(q-1)$)

(5)

Thus, we can achieve the desired relationship if $ed = k\varphi(n) + 1$ (6)

This is equivalent to saying: $ed \equiv 1 \bmod \varphi(n)$, $d \equiv e^{-1} \bmod \varphi(n)$

(7)

That is, e and d are multiplicative inverses $\bmod \varphi(n)$. Note that, according to the rules of modular arithmetic, this is true only if d and e are relatively primesto $\varphi(n)$. Equivalently, $\gcd(\varphi(n), d) = 1$.

We are now ready to state the RSA Algorithm

The ingredients are the following:

- ✓ p,q: two prime numbers (private, chosen)
- ✓ n=p * q (public, calculated)
- ✓ e, with $\gcd(\varphi(n), e) = 1; 1 < e < \varphi(n)$ (public, chosen)
- ✓ $d \equiv e^{-1} \bmod \varphi(n)$ (private, calculated)
- ✓ The private key consists of {d,n}, and the public key consists of {e,n}.

Based on the figure above:Assume that user A has published its public key and that user B wishes to send message M to A. Then B calculates $C = M^e \bmod n$ and transmits C. On receipt of this cipher text, user A decrypts by calculating $M = C^d \bmod n$. It is worthwhile to summarize the justification for this algorithm. We have chosen e and d such that $d \equiv e^{-1} \bmod \varphi(n)$.

3. RESULTS AND EXPERIMENTAL WORK

We have used the Quincy 2005 C++ programming languages how it can encrypt and decrypt the messages from sender to receiver.

The encryption algorithm is done when amessage sent from sender and as well as the decryption algorithm is done at the message delivered to receiver..

I. C++ PROGRAM TO IMPLEMENTATION THE RSA ALGORITHM

```
// C++ Program to Implements the RSA Algorithm: By Desta D.
// The algorthim encrypt and decrypt the
```

Plaintext

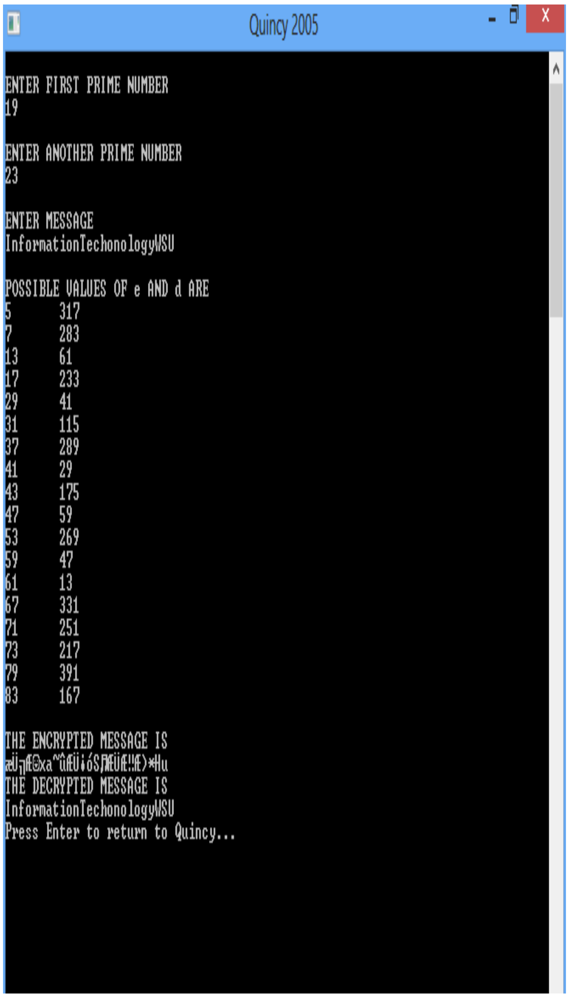
```
#include<iostream>
#include<math.h>
#include<string.h>
#include<stdlib.h>

using namespace std;

long int p, q, n, t, flag, e[100], d[100],
temp[100], j, m[100], en[100], i;
char msg[100];
int prime(long int);
void ce();
long int cd(long int);
void encrypt();
void decrypt();
int prime(long intpr)
{
int i;
   j = sqrt(pr);
   for (i = 2; i <= j; i++)
   {
      if (pr % i == 0)
         return 0;
   }
   return 1;
}
int main()
{
cout<< "\nENTER FIRST PRIME
NUMBER\n";
cin>> p;
   flag = prime(p);
   if (flag == 0)
   {
cout<< "\nWRONG INPUT\n";
      exit(1);
   }
cout<< "\nENTER ANOTHER PRIME
NUMBER\n";
cin>> q;
   flag = prime(q);
   if (flag == 0 || p == q)
   {
cout<< "\nWRONG INPUT\n";
      exit(1);
   }
cout<< "\nENTER MESSAGE\n";
fflush(stdin);
cin>>msg;
   for (i = 0; msg[i] != NULL; i++)
      m[i] = msg[i];
   n = p * q;
   t = (p - 1) * (q - 1);
ce();
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;
}
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()
```

```
{
   long intpt, ct, key = e[0], 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;
cout<< "\nTHE ENCRYPTED MESSAGE
IS\n";
   for (i = 0; en[i] != -1; i++)
printf("%c", en[i]);
}
void decrypt()
{
   long intpt, ct, key = d[0], 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<< "\nTHE DECRYPTED MESSAGE
IS\n";
   for (i = 0; m[i] != -1; i++)
printf("%c", m[i]);
}
```

II. EXECUTED RESULTS BY QUINCY FROM ABOVE CPP CODING



```
ENTER FIRST PRIME NUMBER
19

ENTER ANOTHER PRIME NUMBER
23

ENTER MESSAGE
InformationTechonologyWSU

POSSIBLE VALUES OF e AND d ARE
5       317
7       283
13      61
17      233
29      41
31      115
37      289
41      29
43      175
47      59
53      269
59      47
61      13
67      331
71      251
73      217
79      391
83      167

THE ENCRYPTED MESSAGE IS
æÜ╗Æ☻xa^ûÆÜ¡óSƒÆÜÆ‼Æ)*╫u
THE DECRYPTED MESSAGE IS
InformationTechonologyWSU
Press Enter to return to Quincy...
```

Fig4: Experimental Results in Quincy 2005

4. DISCUSSION

In the above results display the RSA algorithm used to secure the flow of data between clients and server as well as sender and receiver. In data flow between two machines that means source and destination the data should be locked in the two keys that are called Private and public keys. If the data is locked in the key trying the two keys is absolutely difficult for hackers and it strengthening the network security in the organization as well as data security. In this paper we covered how to encrypt and decrypt the flow of data between the source (A) and Destination (B). If the source (A) sends the data to Destination (B) then what task my experiment could automates first A encrypts the Plaintext by using sources(A) private key and

Public keys of A. and sends Cipher text in to Destinations(B). Then B decrypts the messages by using two keys of B which are called public and private keys. The basic thing we want explore in our research is keys interchange between the sources and destinations as it mentioned in above Fig.4-using RSA requires a public key and private key for encrypting and decrypting data over the internet. The main purpose to use such an algorithm is because we need a scalable and secure solution for secure key exchange over the internet. Virtual Private Network(VPN) gateway's as well as other aspects such as secure websites communicating keys across the internet to be used for encrypting and decrypting data could easily be sniffed and stolen by a hacker. For this reason, it is why the public and private key (Asymmetric) mechanism was put into place. So entities could securely agree on a symmetric key over the internet without anyone else being able to capture the secret key [7].

5. CONCLUSION

Securing the flow of data in computer network is the main objectives of Computer Network expert in the any organizations. In this study we identified the RSA algorithm is critical solutions to secure data and locks the flow of data inside the computer networks. As we identified the proposed algorithm has two secured keys which are used to lock and encrypt the data inside the networks the keys are called Private and public keys used in both the sources and destination. We have used the object oriented programming called C++ in Quincy 2005. To automate the RSA algorithms in the real world. Wehave justified by using the executed code in the Quincy 2005, and also we have justified the results and the keys creation, exchange and using in the source and destinations.

REFERENCES

1. Avi K. (2017). Lecture 12: *Public-Key Cryptography and the RSA Algorithm. Computer and Network Security*. page 1-106
2. Besnard, D., &Arief, B. (2004). Computer security impaired by legitimate users. *Computers & Security, 23*, 253-264.
3. Carayon, P., & Smith, M. J. (2000). Work organization and ergonomics. *Applied Ergonomics, 31*, 649-662. Flanagan, J. C. (1954). The critical incident technique. *Psychological Bulletin, 51*(4), 327-358.
4. *Linda P(2008). Introduction to Information Security. Page 1-3*
5. *https://www.techopedia.com/definition/26464*/data-security, accessed **on May 5-6, 2017**
6. *https://www.paloaltonetworks.com/cyberpedia/what-is-network-security,* **accessed on May 29, 2017**
7. *http://www.internet-computer-security.com/VPN-Guide/RSA.html,* Accessed on June 1-3, 2017