International Journal of Information and Communication Technology Research

http://www.esjournals.org

RSA Cryptosystem using Object-Oriented Modeling Technique

N. C. Ashioba , R. E. Yoro
Department of Computer Science Delta State Polytechnic Ogwashi Uku, Delta State, Nigeria

ABSTRACT

Data communication and network security ensures that transmitted data and network are secured from unauthorized users in a communication system. To provide security to the network and data different encryption algorithms have been used. These algorithms have been classified into symmetric and asymmetric key cryptography. This paper presents the RSA cryptosystem using object-oriented model. The RSA is one of the most popular asymmetric key schemes proposed for protection of data and network in a communication system. In this paper we have used the object-oriented model to design and implement the RSA where we used the unified modeling language as the design technique. We implemented the algorithm using object-oriented programming language (C++ programming language). The model enables the senders and receivers in the RSA algorithm have real-world existence where the objects are encapsulated and associated with attributes and methods.

Keywords: *Ciphertext, Cryptography, Decryption, Encryption, Object-Oriented Model, Receiver, RSA and Sender.*

I. INTRODUCTION

Cryptography is one of the methods used to ensure confidentiality and integrity of information in a communication system. It is derived from the Greek word "kryptos" which means secret-writing. Cryptography is the science and art of transforming messages to make them secure and immune to attack [1]. Cryptography is broadly described as the art and science of scrambling data to prevent unauthorized access over unsecured transmission channel. Cryptography basically works on the principal of mathematics that generates different algorithms known as cryptographic algorithm [2]. A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. The cryptographic algorithm works in combination with a key to encrypt the plaintext. Encryption is broadly classified into three categories namely symmetric cryptography, asymmetric encoding and hash function [3]. Symmetric-key cryptography is based on the sender and receiver of messages knowing and using the same secret key. The sender uses the secret key to encrypt the message and the receiver uses the same secret key to decrypt it. The main problem of symmetric key cryptography is getting the sender and receiver to agree on the same secret key without anyone else knowing it. Because all keys in a symmetric key cryptosystem must remain secret, symmetric key cryptography often has difficulty in providing secure key management, especially in open systems with a large number of users. To solve this problem, Diffie and Hellman introduced a new approach to cryptography and, in effect, challenged cryptologists to come up with a cryptographic algorithm that met the requirements for public-key systems. Public-key cryptography is used where each user has a pair of keys, one called the public key and the other private key. Each user's public key is published while the private key is kept secret and thereby the need for the sender and the receiver to share secret information (key) is eliminated. The only requirement is that public keys are associated with the users in a trusted (authenticated) manner using a public key infrastructure (PKI). The public key cryptosystems are the most popular, due to both confidentiality and authentication facilities.

II. RSA Algorithm

The RSA Public Key Encryption algorithm, one of the first public key schemes, was introduced in 1978 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and is named after them as the RSA scheme. The Rivest-Shamir-Adleman (RSA) cryptosystem is one of the best known public key cryptosystems for key exchange or digital signatures or encryption of blocks of data. RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, n, that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in length each, yielding an n with roughly twice as many digits as the prime factors [4]. RSA gets its security from factorization problem. Difficulty of factoring large numbers is the basis of security of RSA. Over 1000 bits long numbers are used.

RSA Problem (RSAP) is also the basis of security of RSA, in addition of factorization problem. The RSA problem assures the security of the RSA encryption and RSA digital signatures. The condition of RSA problem assures that there is exactly one unique m (message) in the field.

The RSA algorithm uses two numbers (e and d) as the public and private keys respectively. In RSA, e and n are announced to the public and d and φ are kept secret. Although, RSA can be used to encrypt and decrypt actual messages, it is very slow if the message is long. Therefore, RSA algorithm is useful for short messages such as a small message digest. Since the algorithm uses two keys for encryption and decryption, the RSA algorithm is considered as an example of asymmetric key

International Journal of Information and Communication Technology Research

http://www.esjournals.org

cryptography. The conceptual design of the RSA algorithm is illustrated in Figure 1.

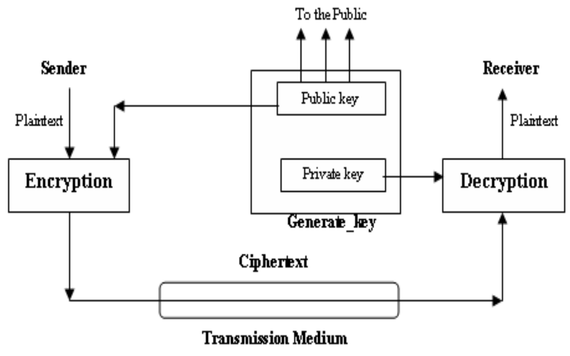

Figure 1: Conceptual design of the RSA Algorithm

RSA algorithm has three subsystems: generation_key, encryption and decryption.

The generate_key subsystem uses two distinct prime numbers as input and it generates the public and private keys as output. The algorithm for the generation_key subsystem is as follows [5]:

1. Choose 2 distinct random prime numbers p and q.
2. Compute n = p.q
3. Compute φ(n) = (p-1)(q-1) {Euler's totient function}
4. Select an integer e, such that 1<e< φ(n) and gcd(e, n) =1
5. Compute d = e^{-1}mod(φ(n))
6. Publish the public encryption key (e,n)
7. Keep the secret private key for decryption (d).

The encryption subsystem uses the public key (e, n) generated by the generation_key subsystem and the plaintext sent by the sender as its input. The encryption subsystem encrypts the message (m) to generate the ciphertext (C) or coded plaintext. The mathematical algorithm for the encryption subsystem is illustrated in (1)

$$C = m^e \bmod n \quad \ldots\ldots(1)$$

The decryption subsystem uses the ciphertext (C) and the private key (d, n) generated by the generate_key subsystem as input. It decrypts the ciphertext to generate the original plaintext. The receiver decrypts the ciphertext by using the mathematical algorithm in (2)

$$m = C^d \bmod n \quad \ldots\ldots(2)$$

III. OBJECT-ORIENTED DESIGN PROCESS MODEL

Object-Oriented Analysis and design (OOAD) model is described [6] as a software engineering approach that model a system as a group of interacting objects that cooperate to solve a task. In Object-Oriented system, each object represents some real-world entity with attributes and operations. Object-oriented model emphasizes the creation of components that encapsulate both data and the algorithms used to manipulate the data [7].

Object-oriented model visualizes the system blueprints by using the unified modeling language. The unified modeling language (UML) is a graphical language for visualizing, specifying, constructing and documenting the artifacts of a software system. The artifacts used in this paper are use-case diagram, class diagram and activity diagram.

Use-case diagram of the RSA Algorithm

The use case diagram is a description of what a system does from a user's standpoint of an external observer [8]. It is a tried-and–true technique for gathering system requirement from user's point of view. It has three major components namely actors, case and relationships. An actor is a person or external system that interacts with the system to achieve a user's goal. Figure 2 illustrates the use-case diagram of the RSA algorithm.

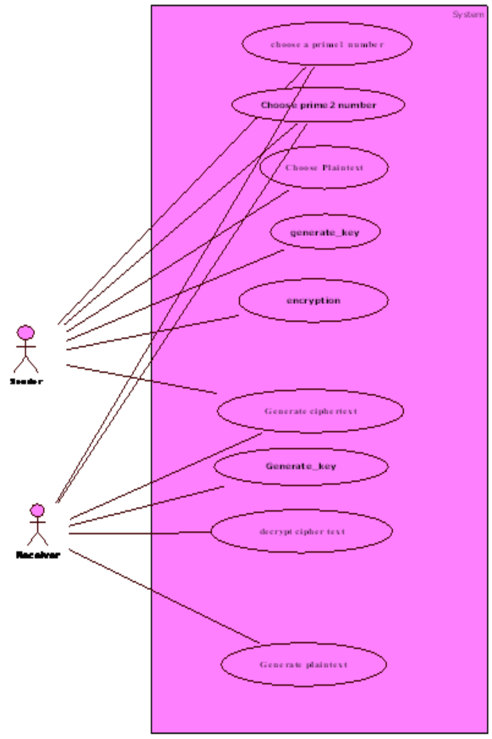

Figure2 Use case diagram of the RSA algorithm

Class Diagram of the proposed system

The class diagram is a static structure that describes the structure of the system by showing the system's classes, their attributes and the relationships between the classes [6]. The class diagram is divided into three sections: class-name, attributes and methods. Others terms used in class diagram

include association, multiplicity, inheritance and visibility. Figure 3 illustrates the class diagram of the RSA algorithm.

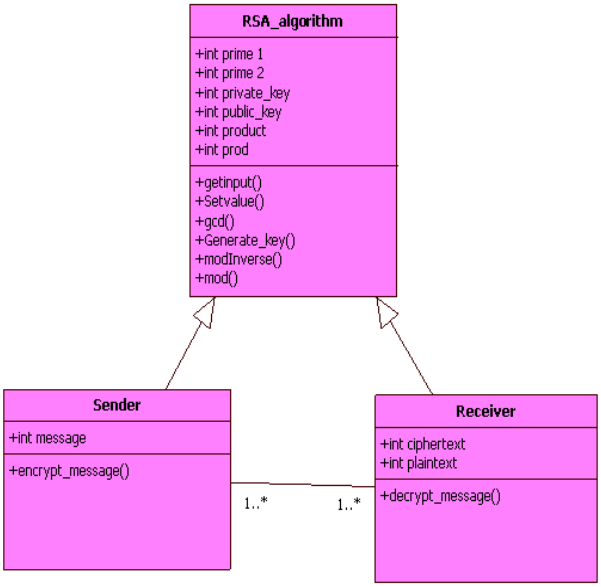

Figure 3 Class diagram of the RSA algorithm

Activity Diagram

An activity diagram is essentially like a flowchart showing the flow of controls from one activity to another. It represents operations on some classes in the system that results to changes in the state of the system. The activity diagram of the RSA algorithm is illustrated in Figure 4.

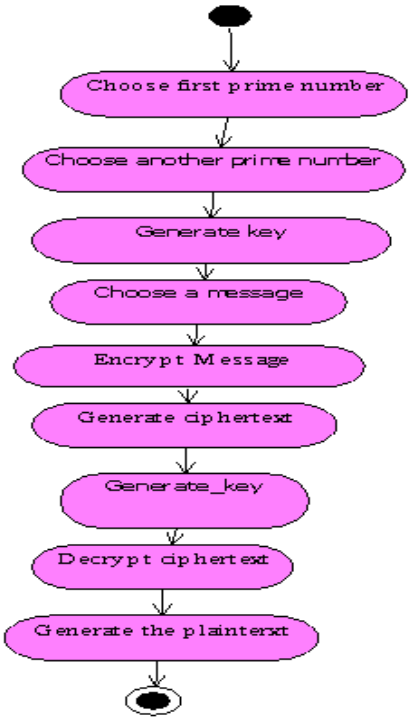

Fig 4: Activity diagram of the proposed system

Object Oriented Programming Language

The object-oriented programming languages adopted in this paper is C++. The language divided the system into three classes, namely, RSA, Sender and Receiver. The sender and receiver are the children of the RSA class. They inherit all properties and methods of the RSA class. The implementation of the RSA algorithm is as follows:

```
#include <iostream.h>
#include <math.h>
#include <stdlib.h>
#include<time.h>
class RSA{
    public:
        int prime1; //The number used by user 1
        int prime2; //The number used by user 2
        int m,n,q;
        int public_key;// public key
        int private_key; // private key
        int ciphertext;
        int ciphertext2;
        int mproduct;
    public:
void getinput()
{
    cout<<"Choose a number e such that gcd(r,q)=1"<<endl;
    cout<<"++++++++++++++++++++++++++++"<<endl;
    cin>> e;
}
void Setvalues(int a, int b)
{
    prime1 = a;
    prime2 = b;
}
int gcd(int m, int n)    // function definition
{                        // block begin
  int  r;                // declaration of remainder
  while (n != 0) {       // not equal
     r = m % n;          // modulus operator
     m = n;              // assignment
     n = r;
```

```
  }                    // end while loop
  return m;            // exit gcd with value m
}
 int modInverse(int a, int m) {
          a %= m;
          for(int x = 1; x < m; x++) {
                   if((a*x) % m == 1) return x;
          }
}
void Generate_key(int a, int b, int f)
{     m = a * b;
mproduct = m;
      q = (a-1)*(b-1);
      public_key = f;
     for (int i=1; i<q; i++)
     {
      if((gcd(f, m)==1) && (gcd(i,q)==1))
      e = modInverse(f,q);
     }
     private_key = e;
      }
int mod(int a, int b, int q)
{
   c = 1;
   for (int i =1; i<=b; i++)
   {
      c = (c*a)% q;
   }
      return c;
}
};
class Sender: public RSA{
      public:
int encryption(int message2)
   {
    Generate_key(prime1, prime2, e);
    ciphertext = mod(message2,public_key,m);
     cout<<"Ciphertext ===  "<<ciphertext<<endl;
    //cout<<"the value of public
key====="<<public_key<<endl;
      cout<<"the value of
message====="<<message2<<endl;
       cout<<"The public key is
"<<"("<<public_key<<";"<<m<<")"<<endl;
      ciphertext2 = ciphertext;
      return ciphertext;
     cout<<"==================="<<endl;
   }
};
class Reciever: public RSA{
     public:
     private:
          int plaintext;
     public:
int decryption(int ciphertext2)
   {
   Generate_key(prime1, prime2, e);
   plaintext = mod(ciphertext2,e,mproduct);
     cout<<"The Private key
is:"<<"("<<private_key<<";"<<m<<")"<<endl;
      cout<<"the value of
ciphertext====="<<ciphertext2<<endl;
     cout<<"The plaintext seen by the receiver
is:"<<plaintext<<endl;
      cout<<"======================"<<endl;
      return plaintext;
        }
};
using namespace std;
int main(int argc, char* argv[])
{
 int message, m;
int prime1; //The number used by user 1
int prime2; //The number used by user 2
Sender user1;
Reciever user2;
 cout<<"Enter the first prime number "<<endl;
 cout<<"++++++++++++++++++++++++"<<endl;
 cin>> prime1;
 cout<<"Enter the second prime number "<<endl;
  cout<<"++++++++++++++++++++++++++"<<endl;
```

```
 cin>> prime2;
 cout<<"Choose a number e such that gcd(r,q)=1"<<endl;
cout<<"++++++++++++++++++++++++"<<endl;
    cin>> e;
    m = prime1 * prime2;
user1.Setvalues(prime1, prime2);
user2.Setvalues(prime1, prime2);
cout<<"The sender information to be sent to the reciever
"<<endl;
cout<<"==============="<<endl;
cout<<"Enter the message to be encrypted "<<endl;
cout<<"==============="<<endl;
cin>>message;
cout<<endl;
cout<<endl;
cout<<"==============="<<endl;
cout<<"The Encryption message for user 1"<<endl;
cout<<"============="<<endl;
int y = user1.encryption(message);
cout<<"==========<<endl<<endl;
cout<<"The decryption result= message for user 2"<<endl;
cout<<"============="<<endl;
user2.decryption(y);
system("Pause");
return 0;
}
```

IV. CONCLUSION

The paper has shown that the RSA algorithm can be model using object-oriented modeling technique. The model enables the senders and receivers in the cryptosystem have real-world existence where their objects were encapsulated and associated with attributes and methods.

Acknowledgment

Our acknowledgments go to our families for their financial ad moral support to the publication of this paper.

REFERENCES

[1]. Forouzan B. A. "Data Communication and Networking (4 Edition)", McGraw Hill Inc. New York, 2008.

[2]. Chaitanya P. and Sree Y. R. "Design of New Security using Symmetric and Asymmetric Cryptography Algorithms." *World Journal of Science and Technology. Vol 2. Issue 10. pp. 83-88, 2012, 2012.*

[3]. Kaushik A. and Satvika, "Extended Diffie-Hellman Algorithm for key Exchange and management" *International Journal of Advances in Engineering Sciences. Vol. 3(3) pp. 67-70, July 2013.*

[4]. Sharma S., Yadav S. J. and Sharma P. Modified RSA Public Key Cryptosystem Using Natural Number Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering. Vol 2. Issue 8. pp. 134-138, 2012.*

[5]. Kumar A., Jakhar S. and Makkar S. "Comparative Analysis between DES and RSA Algorithms". *International Journal of Advanced Research on Computer Science and Software Engineering. Vol 2. Issue 7. pp. 386-391, 2012.*

[6]. Dhiman, K. S., Shatma, A. and Kaur A. "Comprehensive Study of Object Oriented Anaylsis and Design by using the Concept of OOSE" *International Journal of Research in Education Methodology Vol . 1, No. 1. pp. 14-16, 2012.*

[7]. Reddy, R. M., Govindarajulu, P. and Naidu, M. M. "A Process Model for Software Architecture" *International Journal of Computer Science and Network Security, Vol. 7 No. 4, pp. 272-280, 2007.*

[8]. Nayak R., Patheja P. S. and Waoo A. A. "Design of Weather Forecasting System through unified modeling language". *International Journal of Research in Engineering and Applied Sciences Vol. 2 (2) pp. 1189-1194, 2012.*