ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Implementation of Triple RSA

Sapna Sejwani*1, Prof. Prema K.V[2], Mrs. Sarvesh Tanwar[3]
[*1] M.Tech II Year, [2] HOD, [3] Assistant Professor, Department of CSE, Mody University of Science and Technology, Laxmangarh, Rajasthan, India
minuthihu@gmail.com

Abstract

The Rivest Shamir Adleman (RSA) cryptosystem, named after its creators, is one of the most popular public key cryptosystems. It is most widely used for its strong security feature and easy implementation. The RSA cryptosystem has been utilized for many e-commerce applications, various forms of authentication, and virtual private networks in any organizations. The importance of high security and faster implementations paved the way for hardware implementations of the RSA algorithm. This work consists of describing a new approach to enhance RSA security. In this paper we will enhance the security feature by introducing an advance model called Triple RSA. Although RSA has not been attacked yet, it is still prone to attacks. So to enhance its security we have implemented triple RSA just like triple DES which is extremely secure. This model provides along with confidentiality, a strong authentication, data integrity, tamper detection and non repudiation.

.**Keywords**: Authenticity, Confidentiality, Data integrity, Digital signature, Private key, Public key, Public key cryptography, Non Repudiation, Symmetric encryption.

Introduction

Cryptography

Cryptography is where security engineering meets mathematics. It provides us with the tools that underlie most modern security protocols nowadays. It is probably the key enabling technology for protecting distributed systems. It is the art or science encompassing the principles and methods of transforming an intelligible message into one that is unintelligible (cipher text) and then retransforming that message back to its original form (plain text). Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary [7]. Consumer privacy is becoming the most publicized security issue replacing theft and fraud as top concerns in e-commerce [10]. Cryptosystem is system for encrypting and decrypting data. Security of cryptosystem depends on secrecy of the keys rather than the secrecy of the algorithm. It is important to have a large range of possible keys, so that it is not possible to do a "brute force" approach in cracking the algorithm. Traditionally, cryptography was done with just a single key called a secret key, which would have to be known to everyone, and so this was insecure.

- The challenge would be that two parties would have to agree on a secret key without anyone else finding out.
- The secret key method is faster, but less secure.

The public key cryptosystem was introduced in 1976 by Whitfield Diffie and Martin Hellman. It uses public key for encryption, as well as a private key for decryption. Each user gets two keys: one public and one private. The public key is published; the private key is secret. This eliminates the need to share the private key.

Security

The rapid evolution of computing and communication technologies and their standardizations have made the boom in e-commerce possible [1]. The eradication of trust in Internet commerce applications may cause prudent business operators and clients to forgo use of the Internet for now and revert back to traditional methods of doing business [6].

Triple RSA

Triple RSA is an improvement in RSA. The three keys available to a user (his private key, his public key and sender's public key) as shown in

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

figure 1 is utilized in such a fashion that all security features are accomplished [3] [5].

RSA

RSA is public key cryptography algorithm, named after the inventors, Ron Rivest, Adi Shamir, and Len Adleman in 1977. One of the interesting things about RSA is that you can tell anyone about how the encryption works; however, this knowledge is not sufficient to be able to decrypt the cipher text [2] [9] [11]. Only the chosen few who have extra information can decrypt the message.

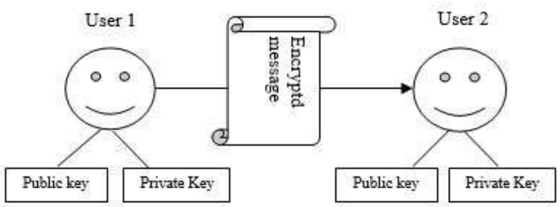

Figure 1: Key availability scenario

Figure 2 depicts the working of RSA encryption and decryption. The sender (BOB) is sending message to receiver (ALICE). The sender will encrypts the message using private key of Alice [8]. So that only Alice can decrypt it as she only has the private key required to decrypt it.

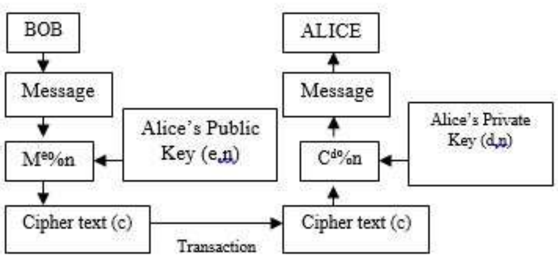

Figure 2: RSA for confidentiality

This is the confidentiality feature provided by the RSA encryption.
RSA can also be used to provide signature [12].

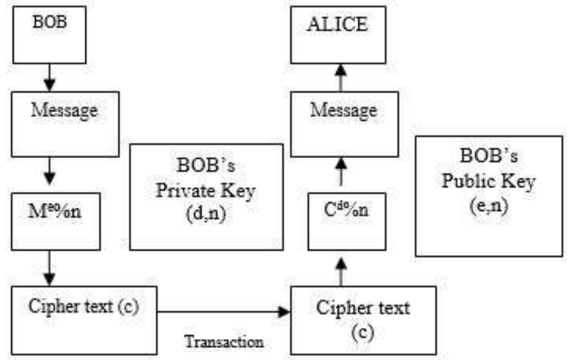

Figure 3: RSA for signature

As shown in Figure 3 BOB encrypts the message using his own private key which is known only to him. Others can decrypt the message using his public key [4] [14].
Types of attacks possible on RSA [13]

- Brute force attack
- Time attack
- Mathematical attack

Triple RSA

Triple RSA is the advancement in the RSA. It provides more security features than RSA. The message is first encrypted using public key of the sender. As the receiver only has the corresponding private key to decrypt, he only can decrypt it. This step provides confidentiality.

Now this encrypted message is further encrypted using sender's private key. This is like signing the message by the sender. As sender only has the private key, it can be decrypted on the receiver's side using sender's public key. Hence it is verified that the message has come from the genuine sender. Sender is authenticated. Also non repudiation is provided. Sender at the end cannot refuse that he sent the message. As the source of the message is him, with his private key the message is encrypted which is available with him only.

At last, the double encrypted is again encrypted using sender's public key. At this point main aim is to provide message integrity. The data can only be decrypted using sender's private key and hence can't be modified while on network. Hence data integrity is also achieved.

Design and implementation

Technologies

- **Front end :** Java Development Tool Kit (version 1.7), Swing, Socket programming
- Back End : Oracle 11g

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Proposed Model

There are two entities who want to communicate among themselves using this new approach as shown in Figure 4. User A sends a message to user B. The message is received with utmost security and confidentiality. Only the authenticated user i.e user B receives the message.

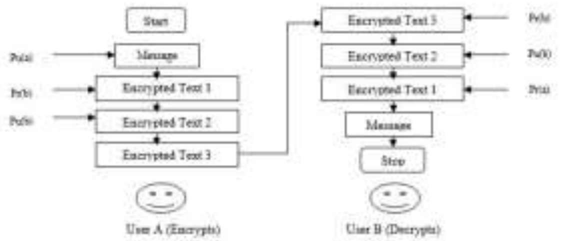

Figure 4: Proposed Triple RSA model

Step 1:

In this proposed model first of all the sender (user A) encrypts the message to be sent to user B using user B's public key. As user B only has the corresponding private key, he is authorized to decrypt the message. So this step provides confidentiality.

E (Message, Pu(B)) = Message1

Where Message: Plaintext (Original message)

Pu(B): Public key of receiver

E: RSA Encryption

Message1: Encrypted message

Step 2:

Next, User A encrypts message1 using his own private key. Only he has his private key so his identity is also validated. Encrypting message by sender's private key is signing the message. This step provides authenticity. While on the receiver side, user B can decrypt it using public key of A.

E (Message1, Pr(A)) = Message2

E (Message2, Pu(B)) = Message3

Where Messag2: Double encrypted message

Pu(B): Public key of receiver

E: RSA Encryption

Where Messag1: Encrypted message

Pr(A): Private key of sender

E: RSA Encryption

Message2: Double encrypted message

Step 3:

Finally, User A encrypts message1 using receiver's public key. Only receiver has his private key so user B only can decrypt the message. This provides enhanced security on the message transaction. The data in the message can't be tempered as security has been improved. This step provides message integrity.

Message3: Triple encrypted message

3.2 Code

(a) Unique random number P and Q generation

```
randomNumber = random.nextInt(max - min) + min;
for(a1=2;a1<=randomNumber;a1++)
{
        if(randomNumber%a1==0)
        {
        break;
        }
}
```

(b) Checking if two numbers are distinct and generation of n

```
if(a1==randomNumber)
{
c++;
if(c==1)
    {
    p=a1;
    }
else if(c==2&&p!=a1)
    {
    q=a1;
    break;
    }
}
n=p*q;
```

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

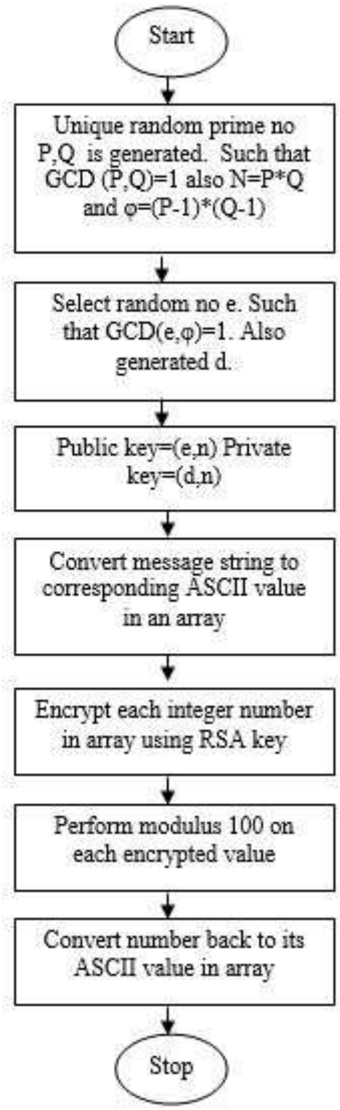

Figure 5: Working model of triple RSA

(c) Generating e and d (public and private keys respevtively)

```
randomNumber = random.nextInt(phi - 1) + 1;
gcd= new GCD().gcdCal(phi,randomNumber);
if(gcd==1)
{
en_key=randomNumber;
b_1=new BigInteger(en_key+"");
b_2=new BigInteger(phi+"");
b_3=b_1.modInverse(b_2);
String str= b_3+"";
d_key = Integer.parseInt(str);
break;
}
```

(d) Converting String into ASCII

```
b = str.toCharArray();
a=new int[b.length];
for(i=0;i<b.length;i++)
{
a[i]=b[i];
}
```

(e) Encrypting message

```
expo=new BigInteger(en_key+"");
b2=new BigInteger(n+"");
for(i=0;i<a.length;i++)
    {
    b1=new BigInteger(a[i]+"");
    b3 = b1.modPow(expo,b2);
    String str1= b3+"";
    a[i] = Integer.parseInt(str1);
    show[i]=a[i]%100;
    enc_show[i]=(char)show[i];
    }
```

(f) Decrypting message

```
expo1=new BigInteger(d_key+"");
 b12=new BigInteger(n+"");
 dec_pass=new char[a.length];
for(i=0;i<a.length;i++)
 {
 b11=new BigInteger(a[i]+"");
 b13 = b11.modPow(expo1,b12);
 String str11= b13+"";
 a_dec[i] = Integer.parseInt(str11);
 dec_pass[i]=(char)a_dec[i];
 }
```

(g) Converting ASCII into string

```
decpass=decpass+dec_pass[i];
dec_show[i]=(char)a_dec[i];
```

Implementation

(a) On sender side (Encryption)

Step 1: Sender encrypts the original message by receiver's public key as shown in Figure 6. (Confidentiality)

E (Message, Pu(B)) = Message1

Where

Message: Plaintext (Original message)
Pu(B): Public key of receiver
E: RSA Encryption
Message1: Encrypted message

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Figure 6: First encryption

Step 2: Sender encrypts the encrypted message by sender's private key as shown in Figure 7. (Authenticity)

E (Message1, Pr(A)) = Message2

Where

Messag1: Encrypted message
Pr(A): Private key of sender
E: RSA Encryption
Message2: Double encrypted message

Figure 7: Second encryption

Step 3: Sender encrypts the double encrypted message by receiver's public key as shown in Figure 8.(Message integrity)

E (Message2, Pu(B)) = Message3

Where

Messag2: Double encrypted message
Pu(B): Public key of receiver
E: RSA Encryption
Message3: Triple encrypted message

Figure 8: Third encryption

(b) On receiver side (Decryption)

Step 1: Receiver decrypts the triple encrypted message by his private key as shown in Figure 9.

D (Message3, Pr(B)) = Message2

Where

Message3: Triple encrypted message
Pr(B): Private key of receiver
D: RSA Decryption
Message2: Double encrypted message

Figure 9: First decryption

Step 2: Receiver decrypts the double encrypted message by sender's public key as shown in Figure10.

D (Message2, Pu(A)) = Message1

Where

Message2: Double encrypted message
Pu(A): Public key of sender
D: RSA Decryption
Message1: Encrypted message

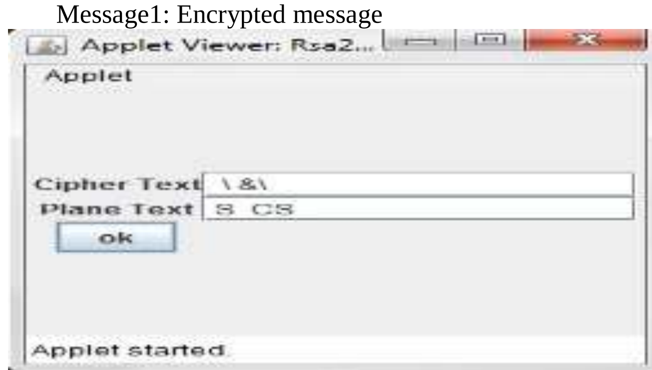

Figure 10: Second decryption

Step 3: Receiver decrypts the encrypted message by his private key as shown in Figure 11.

D (Message1, Pr(B)) = Message

Where

Message1: Encrypted message
Pr(B): Private key of receiver
D: RSA Decryption
Message: Plaintext (Original message)

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

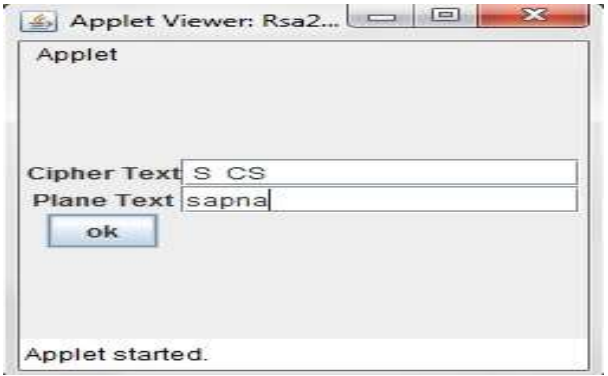

Figure 11: Third decryption

Results

Here we have a comparison table Table 1 for security features like confidentiality, data integrity and signature.

Security Feature	**Conventional Algorithm**	**Our Algorithm**	**Time taken by Conventional Algorithm**	**Time taken by our Algorithm**
Confidentiality	Simple RSA	Triple RSA (Step 1)	35 ns	35ns
Integrity	SHA1	Triple RSA (Step 2)	77ns	35ns
Signature	Simple RSA	Triple RSA (Step 3)	35ns	35ns
Total Time			**147 ns**	**105ns**

Table 1: Comparison of Triple RSA with other algorithms

Integrity is provided by some digest algorithm like MD5 or SHA. To show integrity with RSA we use SHA1. As RSA can

Only provide confidentiality and authentication. It can't provide message integrity. But Triple RSA can provide message integrity.

The experiment was implemented on

Processor: Intel® Core™ Duo CPU P8600 @ 2.40 GHz.

RAM (Internal memory required): 4 GB

Operating system: Windows 7 Ultimate

System Type: 64 bit operating system

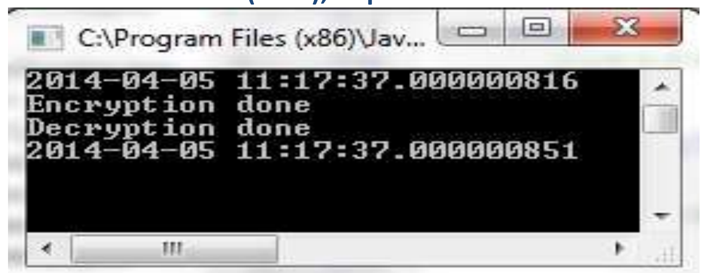

Figure 12: Time taken by RSA (35 ns)

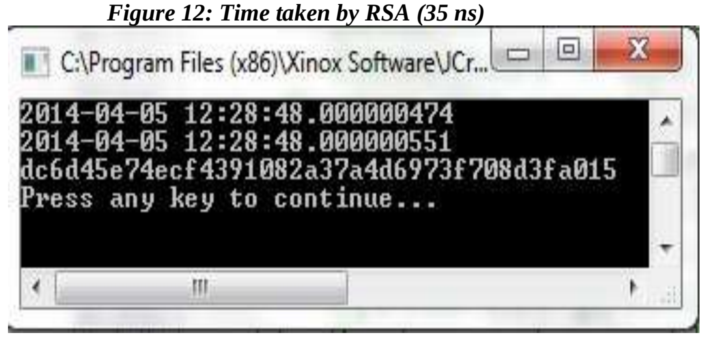

Figure 13: Time taken by SHA1 (77 ns)

Hence it is proved that Triple RSA provides all the three security features: Confidentiality, Message Integrity and Authentication (via Digital Signature) in less time i.e. 105 ns compared to other algorithms which take 147 ns.

Conclusion

The given approach of cryptosystem is targeting all the required security features. It is an efficient way to have any transaction in open network. This approach can be merged with any online transaction to provide strong authentication, privacy, confidentiality, data integrity, tamper detection and non repudiation. This approach covers all the security aspects in just three steps.

References

[1] Niranjanamurthy, M., and DR Dharmendra Chahar. "The study of E-Commerce Security Issues and Solutions." perspectives 2.7 (2013).

[2] Cohen, Aaron E., and Keshab K. Parhi. "Architecture optimizations for the RSA public key cryptosystem: a tutorial." Circuits and Systems Magazine, IEEE11.4(2011): 24-34.

[3] Sun, Hung-Min, et al. "Dual RSA and its security analysis." Information Theory, IEEE Transactions on 53.8 (2007): 2922-2933.

[4] Geiselmann, Willi, and Rainer Steinwandt. "Special-purpose hardware in

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

cryptanalysis: The case of 1,024-Bit RSA." Security & Privacy, IEEE 5.1 (2007): 63-66.

[5] *Chang, Weng-Long, Minyi Guo, and Michael Shan-Hui Ho. "Fast parallel molecular algorithms for DNA-based computation: factoring integers."NanoBioscience, IEEE Transactions on 4.2 (2005): 149-163.*

[6] *Bo, Yang, Mao Jane, and Zhu Shenglin. "A Traitor Tracing Scheme Based on the RSA Scheme." TENCON 2005 2005 IEEE Region 10. IEEE, 2005.*

[7] *Sengupta, A., C. Mazumdar, and M. S. Barik. "e-Commerce security—A life cycle approach." Sadhana 30.2-3 (2005): 119-140.*

[8] *Kim, Ho Won, and Sunggu Lee. "Design and implementation of a private and public key crypto processor and its application to a security system."Consumer Electronics, IEEE Transactions on 50.1 (2004): 214-224.*

[9] *Hong, Jin-Hua, and Cheng-Wen Wu. "Cellular-array modular multiplier for fast RSA public-key cryptosystem based on modified Booth's algorithm." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 11.3 (2003): 474-484.*

[10] *Marchany, Randy C., and Joseph G. Tront. "E-commerce security issues."System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on. IEEE, 2002.*

[11] *Su, Chih-Yuang, et al. "An improved Montgomery's algorithm for high-speed RSA public-key cryptosystem." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 7.2 (1999): 280-284.*

[12] *Gennaro, Rosario, Hugo Krawczyk, and Tal Rabin. "RSA-based undeniable signatures." Advances in Cryptology—CRYPTO'97. Springer Berlin Heidelberg, 1997. 132-149.*

[13] *Kocher, Paul C. "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems." Advances in Cryptology—CRYPTO'96. Springer Berlin Heidelberg, 1996.*

[14] *Iwamura, Keiichi, Tsutomu Matsumoto, and Hideki Imai. "High-speed implementation methods for RSA scheme." Advances in Cryptology—EUROCRYPT'92. Springer Berlin Heidelberg, 1993.*