South Asian Journal of Engineering and Technology Vol.2, No.17 (2016) 67–72

ISSN No: 2454-9614

A Novel Approach to Secure Data using Homomorphic Encryption

A. Saranyadevi, s. Anguraj*

[a)] *Department Of Computer Science and Engineering, K.S.R.College of Engineering, Namakkal -, Tamilnadu, India.*

*Corresponding Author: A. Saranyadevi

E-mail: saranyaksr92@gmail.com,

Received: 09/11/2015, Revised: 13/12/2015 and Accepted: 12/03/2016

Abstract

Overwhelm word of Conventional Encryption techniques plays a dramatic role in securing the data and there are many algorithms that provide protective sensitive data on various applications. The objective of all these techniques enables users to get a high secure and authentication process of efficiency across data transmission. Even so, these conventional encryption techniques are vulnerable to various attacks. In addition, the normal regular encryption techniques fail occasionally during the decryption process while providing the authentication or authorization services. Hence to overcome these attacks, new proposed encryption techniques and issues based on homomorphism properties are used. This paper gives a novel approaches on security and finally authentication processes are improved at the decryption side.

Keywords: Homomorphic Encryption, Bit/Byte Fault RSA e^{th} root Algorithm.

Reviewed by* **ICETSET'16 *organizing committee*

1. Introduction

The advancements in the field of Information Security and data embedding has made several modifications in the society as a whole which involves how a secure transaction takes place to communicate with one another. When any piece of information is shared using internet, it requires high level of security. As the information is shared among various users, it is mandatory that the information remains unchanged.

For last few decades the demand to keep data secret with the corresponding algorithm increased rapidly over many technologies. The key size (x) ranges high for the bit value and the process of encryption over the transmission also prepared as it is difficult to increase the internal consistency and reduce corruption over electronic data. **[1][2]** On a whole, this includes the increased growth field level in both the network security and the securing techniques together with the several fashionable ideas such as networks can be secured by encryption and also networks can be secured by firewalls.

The three main securities that are used to keep the information intact and safe from any unauthorized access are confidentiality, authenticity and availability. Confidentiality acts as a basic requirement during any data transfer as it is required to keep any disclosed information as a secret. Authentication can be used to prevent an attacker altering the contents of the information and also keeps them from replacing a fake document for a legitimate one. Availability is an important component among the three because if the resource is unavailable, it proves as much of a threat to the organization as the information being modified or swapped.

Passive attacks are those when the attacker just tries to monitor the information rather than making any changes to it. The attacks can prove dangerous as they cannot be detected easily and the intruder may disclose the information to the public without any consent. But passive attacks are easier to avoid, encrypting the data before transmitting can prevent this kind of attack. An active attack unlike passive attack is very difficult to prevent but easy to identify. Here, the intruder makes changes to the data or complete swap it with a fake.

The algorithms provide a textured way of performing the encryption or decryption which may be a great asset when facing against such active attacks. As decryption usually involves a same format and just can be broken down into the original information. Hence, the technique moves to homomorphic cryptosystem. As a part, these techniques suffer a greater attack on sensitive data on transmission such as destruction of message, tracking of scheme and theft of data by an eavesdropper **[2].** Those attacks involve the combinations of vulnerabilities such as stack overflow attacks (where you pass an over-long parameter to a program that carelessly executes part of it) and password guessing, both of which were used by Internet worm. Not only do these attacks extends to vulnerabilities but also on another progress such as Attacks on Local Networks (i.e., these include Reimage attack, Boomerang attack, Brute-force attack, Stream cipher attack and so on) and Attacks Using the Internet Protocols and Mechanisms (i.e., SYN Flooding, Smurfing, Distributed Denial-of-service Attacks, Spam and Address Forgery and Spoofing Attacks). To provide a high constrain on data, encryption is made along with authentication and confidential secrecy in this proposed system.

The security of the method should not rely on the obfuscation of their codes, but it should be on the key method used in the corresponding techniques. So, the solution to this problem is to make a difficult level of key usage in different techniques with the high range of encrypted data.(RE) This gazed approach on encrypted data can be processed back at least to the method of Rivest, Adleman and Dertouzos[RAD78] under the name of “homomorphism."

2. Homomorphic Cryptosystems

A homomorphism is a technique of same or similar form computation using mathematical formulae with an ability to progress over cipher text instead of plaintext **[7].** That is, the process of client starts with the data x and implies the encryption Enc(x) to the server. On other hand, the process of server takes the cipher text Enc(x) and evaluates a function f with the extended value of x giving decrypted result Enc(f(x)).If necessary; the client can get

back the data by achieving the wanted functionality. It can be implemented as.

$$\forall\, m_1, m_2 \in M,\ E(m_1 \bullet_M m_2) \leftarrow E(m_1) \bullet_{\varsigma} E(m_2)$$

From above arithmetic equation, let's compute additive homomorphism using addition operator and multiplicative homomorphism using multiplication operator. This process as a whole provides an algebraic homomorphism using numerical operation (i.e.,) it performs the operation on the plaintext before encryption or on focused cipher text after encryption, these prove to be a kind of secret headway and creates an adversary security on plaintext. Mainly, to overcome this entrusted with privacy information, a classified secrecy is made with the RSA multiplicative homomorphism.

3. RSA Homomorphism

RSA (Rivest, Adleman and Dertouzos, 1978) homomorphism is a rich technique in providing a best security level [3][4]. It furthermore, provides a basic homomorphic operation of multiplication modulo n. The definition of RSA cryptosystem can be given as,

Let n = p.q where p and q are primes. Taking the value of a and b such that $ab \equiv 1 \pmod{\varphi(n)}$ where $\varphi(n) = (p - 1) * (q - 1)$. In this process n and b is public while p, q and a are private. Normal computation of RSA cryptosystem is $c_1 = Enc(m1) = m_1^e \bmod n$ and $c_2 = Enc(m2) = m_2^e \bmod n$. Taking this two cipher text as input plaintext for computation then the RSA homomorphism **[5] [6].**

$$c_1 c_2 \bmod n = (m_1^e \bmod n) * (m_2^e \bmod n)$$
$$= (m_1^e . m_2^e) \bmod n$$
$$= (m_1 . m_2)^e \bmod n$$

4. Module Description

The Cryptography techniques are used to handle the secured data transmission process. The system protects the homomorphic RSA from bit and byte fault based attacks.

The system is divided into four major modules. They are Data Selection, Bit Faults and Byte Faults based Protection, Encryption Process, and Decryption Process.

File selection and key generation operations are carried out under the data selection process. Bit fault based protection is used to secure the RSA algorithm. The RSA security is enhanced with byte fault based protection process. Data encrypting is carried out under the data encryption process. And that encrypted data is sent to the receiver. Data decryption module is designed to decrypt the data values.

5. Data Selection

The data selection operation is performed in the sender application. The secret text file is selected by the user for the transmission process. The key generation process creates the public key and secret key values for the RSA algorithm.

6. Bit/Byte Faults Based Protection

The bit insertion attacks detection is performed in the Concurrent Error Detection (CED) model. The bit padding scheme is used in the bit fault handling process. The bit padding process is initiated before the encryption process. Signature integration is used to secure the RSA model. The byte fault error handling scheme is used to protect byte insertion attacks. The signature attachment process is applied in the encryption process. The interactive CRT algorithm is used to control the byte faults' errors. The encrypted data values are passed into the encrypting process.

The system is designed to handle precise bit fault and precise byte fault models. The system is also aimed to reduce the memory overhead and detection latency. The system is designed to detect accidental and intentional faults. Chinese remainder theorem (CRT)-RSA algorithm, Giraud's algorithm and vigilant algorithm are used to analyze latency and resource levels.

The adversary has precise control on both timing and location. This attack can be modeled as an addition or subtraction of a single bit. Pre-computational steps are integrated with the CRT-RSA model. RSA signature is added to the scheme. The attack is initiated by inserting unknown error byte at a known position. The number of bits affected can only be bounded by a block of few bits (byte). Encryption signatures are used with the system. Infective CRT-RSA algorithm is adapted to solving fault errors.

Hardware failures initiate the accidental faults. Content modifications are performed in intentional faults. Memory encryption/decryption techniques are used. Addresses scrambling methods are applied. Processing time and memory resources are limited in the error detection process. Memory requirement is increased for intermediate results. Parallel execution model is integrated with the system. Randomized bit paddling is controlled concerning the memory availability.

6.1 Advantages

- ✓ Latency period is minimized
- ✓ Efficient resource utilization
- ✓ Security without fault compromised

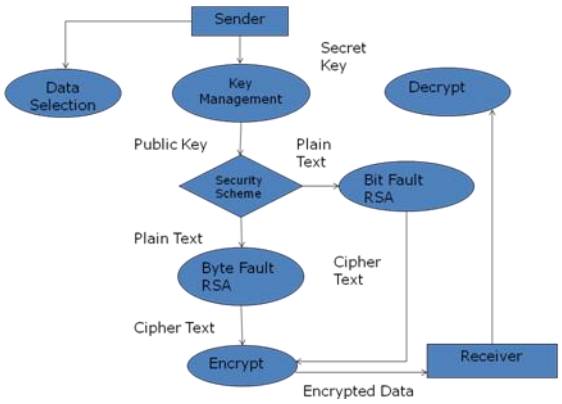

Fig 2 Bit/Byte Fault Based RSA e^{th} root Algorithm

7. Encryption Process

The RSA cryptosystem is the public-key cryptosystem which is used for the secure communication of data. The homomorphism provides a basic homomorphism operation of multiplication modulo n.

Let p=3 and q=11 where p, q should be a selected in asymmetric form. Formulate n=p*q=3*11=33.Also calculate the value of (n) = (p-1)*(q-1) = 2*10=20.

Randomly, assign the value for e such that 1<e<(n) and e & n are co-prime(i.e., where 1<e<(n), gcd(e,(n))=1).Let e=7 compute a value for d such that (d*e)%(n)=1.

One elucidation is d=3[(3*7)%20=1] which can be derived from Euclid Algorithm that solves following equation to find decryption key d.

$$ed = 1 \bmod (n) \text{ and } 0<d<n$$

According to RSA homomorphism, public encryption key:= {e,n} and keep secret private decryption key:={d,p,q}.

- Public key is (e,n)=(7,33) and private key is (d,p,q)=(3,3,11)
- Encryption $Enc(m_1)=m_1^e \bmod n$ and Decryption $Dec(c_1)=m_1^d \bmod n$

7.1 Encryption Process

The encrypted cipher text is further encrypted in RSA e^{th} root algorithm using the formula that $c_1c_2= ((m_1 +1) (m_2 +1))^e \bmod n$. And $(m_1 +1)$ is used as a message value for encryption process. Digital signatures reused for the data integrity verification process and then transfer the encrypted data to the destination node. The data values are received and updated by the receiver node. Secure Hash Algorithm (SHA) is used for the data verification process.

7.2 Decryption Process

Data decrypt operations are carried out under the decryption process. The cipher text is extracted from the log files and decrypts the data in RSA e^{th} root algorithm using the formula that $m_1m_2= ((c_1-1)(c_2-1))^d \bmod n$. Instead of c; decryption process takes the message value as (c-1). When the cipher text during transmission is hacked by eavesdropper and processed the techniques, they can't find the message easy until we share the key usage of $(m_1 - 1)$. Data's decryption process is carried out with bit/byte fault elimination mechanism. Data decryption process is carried out with bit/byte fault elimination mechanism.

8. Conclusion

The security without authentication and Integrity leads to failure of existing method. Hence In this paper we have described how the data is transmitted across the internet and how we can benefit from it and use it in many applications. As every technology has a flaw and need of securing data application, Security is a chief anxiety in the entire field for data that is stored in a cloud and it becomes very tough to achieve operations on the encrypted data, hence we can use homomorphic encryption to secure our data and also perform tasks on it. The whole systems are processed for computing better operation and authentication process.

References

[1] EktaChauhan, Sonia Vatta, International Journal of Advanced Research in Computer Science and Software Engineering, Cyber Security in Data Mining Using Homomorphic Encryption Volume 3, Issue 6, June 2013 ISSN: 2277 128X

[2] ParsiKalpana, Ravindran et al., Data Storage Security Using Partially Homomorphic Encryption in a Cloud, International Journal of Advanced Research in Computer Science and Software Engineering 3(4),April - 2013, pp. 603-606.

[3] Rajan.S.Jamgekar, GeetaShantanu Joshi , File Encryption and Decryption Using Secure RSA, International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-1, Issue-4, February 2013.

[4] Shilpa M Pund, Chitra G Desai, Implementation of RSA algorithm Using Mersenne Prime, International Journal of Networking & Parallel Computing www.cirworld.com (ISSN: 2319-4529) Volume 1, Issue 3, Dec 2012-Jan 2013.

[5] B.Persis Urbana Ivy, PurshotamMandiwa.MukeshKumar , A modified RSA cryptosystem based on 'n' prime numbers , International Journal Of Engineering And Computer Science ISSN:2319-7242, Volume1 Issue 2 Nov 2012 Page No. 63-66.

[6] Samoud Ali, CherifAdnen, RSA ALGORITHM IMPLEMENTATION FOR CIPHERING MEDICAL IMAGING, Samoud Ali, et al International Journal of Computer and Electronics Research [Volume 1, Issue 2, August 2012] ISSN: 2278-5795.

[7] Taher ElGamal. *A public key cryptosystem and a signature scheme based on discrete logarithms*. In G. R. Blakley and David Chaum, editors, CRYPTO 1984, volume 196 of Lecture Notes in Computer Science, pages 10–18.Springer, 1984.

[8] Jasleen Kour, Deepankar Verma, International Journal of Engineering Research in Management & Technology ISSN : 2278-9359 (Volume-3, Issue-5).