V.Saravanakumar / International Journal of Engineering Research and Applications (IJERA)
ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 4, Jul-Aug 2013, pp. 508-510

ERSA: Secure and Enhanced RSA

V.Saravanakumar
Assistant Professor, Department of Computer Science and Engineering, Kongunadu College of Engineering and Technology, Tamil nadu, India

Abstract

This paper is devoted to the analysis of various cryptanalysis attack .This cryptanalysis attack mainly happens on the encrypted message which is to be passed over communication channel. The cryptanalysis attack is used to get the key from the encrypted message. This process uses public key cryptography via RSA algorithm with some modification. There are two keys used in RSA algorithm for effectiveness in the aspect of security one being the public key used for all and the other being the secret key. This paper presents the analysis of the security of information with enhanced the speed of encryption and decryption process.

Index Terms— Public key Cryptography, Encryption, Decryption, RSA algorithm, Cryptanalysis, Public Key Infrastructure

I. Introduction

1.1 Cryptography

As long as there are creatures endowed with language there will be the desire for confidential communication -- messages intended for a limited audience. Governments, companies and individuals have a need to send or store information in such a way that on the intended recipient is able to read it. Generals send orders, banks send fund transfers and individuals make purchases using credit cards.

Cryptography is an indispensable tool for protecting information in computer systems [2]

Cryptographic methods involve two basic activities: hiding information from unauthorized parties and making information unintelligible to individuals other than the intended recipient(s). There are two types of key-based encryption: *symmetric* (shared key) and *asymmetric* (public key) encryption.

1.2 Symmetric Vs Asymmetric Algorithms

Symmetric algorithms use the same key for encryption and decryption (analogous to a key that locks and unlocks a door), while asymmetric algorithms use a different key for encryption and decryption (analogous to a key that locks a door, but requires a different key to unlock the same door).

1.3 Public key cryptography

Public-Key Cryptography was first suggested in 1976 by Diffie and Hellman and a public-key cryptosystem is one which has the property that someone who knows only how [to] encipher ('disguise') a piece of information CANNOT use the enciphering key to find the deciphering key without a prohibitively lengthy computation. This means that the information necessary to send private or secret messages, the enciphering algorithm along with the enciphering key, can be made public-knowledge by submitting them to a public directory.

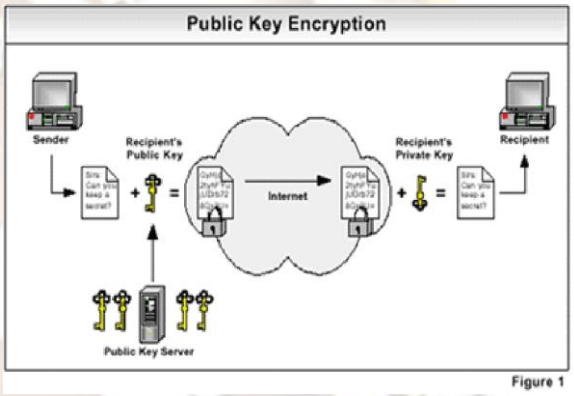

Figure 1

A public key is a value provided by some designated authority as an encryption key that, combined with a private key derived from the public key, can be used to effectively encrypt messages and digital signatures. The use of combined public and private keys is known as *asymmetric* cryptography. A system for using public keys is called a public key infrastructure (PKI).

II. RSA Algorithm

The first public-key cryptosystem, the RSA Algorithm, was developed by Ronald Rivest, Adi Shamir and Leonard Adleman at MIT in 1977. RSA authentication is known to be the most widely adopted encryption method implemented in servers, media players, smart phones and similar devices apart from serving as a security system for safeguarding sensitive information online.

There are two keys used in RSA algorithm for effectiveness in the aspect of security one being the public key used for all and the other being the

secret key. The RSA algorithm involves three steps: key generation, encryption and decryption.

2.1 Key Generation

RSA involves a **public key** and a **private key.** The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way:

1. Select two distinct prime numbers p and q *randomly*.
2. Compute $n = p*q$.
3. Compute $\varphi(n) = (p-1)(q-1)$
4. Choose an integer e such that $1 < e < \varphi(n)$ and GCD $(e, \varphi(n)) = 1$.
5. Calculate $d = e^{-1} \mod \varphi(n)$
6. Public Key=ku= {e, n)
7. Private Key=Kr= {d, n}

2.2 Encryption

Plain text: M<n
Cipher text: C= $M^e \pmod n$.

2.2 Decryption

Cipher text: C
Plain text: M=$C^d \pmod n$.

2.3 Problems in RSA

1. The alphabets in the plain text are represented by numbers ranging from 1 to 26. Hence it is not possible to represent the special; characters in the plain text.
2. The encryption and decryption technique is applied to each and every character of a text, even though the character is repeated in the plain text. So it requires more time for the redundant calculation.
3. As the calculation result is repeated in the cipher text, it will be ease for the intruders to identify corresponding characters, thus by trapping the message in a easier way, which results in security lapses.

Solution:

The above mentioned problems are being resolved by the following manner:

1. Instead of numbers, the characters in the plain text are represented by UNICODE-32 which can be able to represent the special characters;
2. The second one is to be solved by the back substitution. Once the result is computer, it cannot be repeated again. For this purpose, the results are maintained and will be reused in upcoming calculation.
3. In case of numbers in cipher text, the intruders may able to detect the characters easily. So the symbols are also used in cipher text representation the symbol is substitute exactly once with a particular number. For the next occurrence, some other symbols are used to represent the same number. Finally the entire cipher text is in the form of symbols instead of numbers.

III. Enhanced RSA Algorithm (ERSA)

The main intention of this algorithm is to improve the security and to eliminate the redundant calculation in encryption and decryption. This algorithm also uses the same methodology for the secret and public key generation. It also eliminates the redundancy by using the previously calculated result, if the character is repeated.

3.1 Key Generation

The ERSA algorithm provides security for the data, which is transferred in a network. The algorithm consists of the following steps

1. Determine public and secret key. The computation includes the following steps.
   a) Select two distinct prime numbers p and q *randomnly*.
   b) Compute $n = p*q$.
   c) Compute $\varphi(n) = (p-1)(q-1)$
   d) Choose an integer e such that $1 < e < \varphi(n)$ and
   e) GCD $(e, \varphi(n)) = 1$.
   f) Calculate $d = e^{-1} \mod \varphi(n)$
   g) Public Key=ku= {e, n)
   h) Private Key=Kr= {d, n}
2. The plain text is represented using UNICODE-32.

3.2 Encryption

In encryption, the plain text is converted to the cipher text by using this formula if the character is not already exists in the text.
Plain text: M<n
Cipher text: C= $M^e \pmod n$.

3.3 Decryption

In decryption, the cipher text is converted into UNICODE-32 by using this formula if the character is not already exists in the text.
Cipher text: C
Plain text: M=$C^d \pmod n$.

3.4 The UNICODE-32 is finally converted into plain text.

IV. Result

The observed results are tabulated in table

File Size (Bytes)	Algorithms	Encryption Time (Sec)	Decryption Time (sec)	Execution Time (sec)
298	RSA	23	3	26
	ERSA	20	2	22
998	RSA	65	7	72
	ERSA	62	6	68
2095	RSA	296	14	310
	ERSA	286	10	296
3278	RSA	397	17	414
	ERSA	358	13	371

Table 1

The above values are represented in the graph given in figure 2.

The file size is taken to be in X axis and the execution time in Y axis. The execution times of all the algorithms are represented simultaneously in chart.

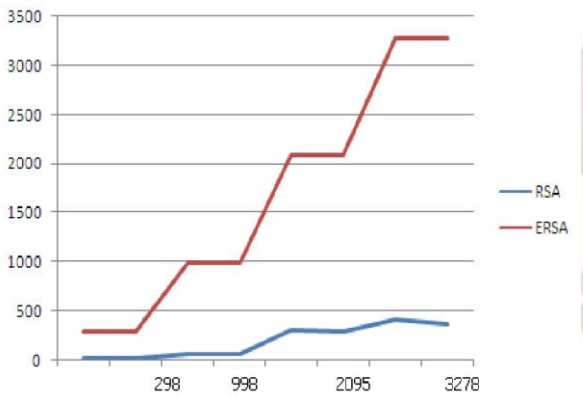

Figure 2

V. Conclusion

The algorithms RSA and ERSAare compared based on their execution time. The same set of data is applied to test the speed of each algorithm. Since in all the test cases, the value of encryption key is larger than the value of decryption key. The encryption time is comparatively higher than the decryption time. In RSA, the execution time primarily depends on the file size. As the file size increases, the execution time will be increased automatically. In ERSA, the execution time is purely based on the repetition of each character in the test data. If the number of the occurrences of a character increases (as the same is being represented by the same number) the execution time will get reduced with in turn enhance the performance of the system. Even for an increased file size the execution time may remain same due to the duplication of characters. In ERSA the speed depends on the file size. In this algorithm, the encrypted message is again encrypted; the cipher text is represented by the way of symbols instead of numbers. Thus the new algorithm introduced to support both the speed and security is known as ERSA algorithm.

VI. Reference

[1] Aboud, S.J.; Al-Fayoumi, M.A.; Al-Fayoumi, M.; Jabbar, H.;" **An Efficient RSA Public Key Encryption Scheme",** *Fifth International Conference on Information Technology: New Generations, 2008. ITNG 2008.*

[2] Harn, L.; Huang, D.; " **A protocol for establishing secure communication channels in a large network ",** *IEEE Transactions on Knowledge and Data Engineering"*, 2007.

[3] Lin Zi; Shi WenXiao; Wang Li;" **A study and analysis on a high intensity public data encryption algorithm",** *Proceedings of the 3rd World Congress on Intelligent Control and Automation, 2000.*

[4] www.RSA.org

[5] Cryptography and Network Security, Third Edition,William stallings,2007.