File Encryption and Decryption using Secure RSA

Rajan. S. Jamgekar, Geeta Shantanu Joshi

Abstract-In this paper we have introduced secure RSA for secure file transmission. There are many cases where we need secure file transmission for example in banking transactions, e-shopping etc [4]. In this paper we present modified RSA algorithm for secure file transmission. RSA algorithm is asymmetric key cryptography also called Public Key cryptography. Two keys are generated in RSA, one key is used for encryption &other key which is only known to authenticated receiver can decrypt message. No other key can decrypt the message. Every communicating party needs just a key pair for communicating with any number of other communicating parties. Once someone obtains a key pair, he /she can communicate with anyone else. RSA is a well known public key cryptography algorithm and was one of the first great advances in public key cryptography. Even if it is efficient algorithm it is vulnerable to attackers. With the help of all brute force attacks hacker can obtain private key.Many improvements has been done to improve RSA like BATCH RSA, MultiPrime RSA, MultiPower RSA, Rebalanced RSA, RPrime RSA etc.

As craze of internet is increasing exponentially, it is used for email, chatting, transferring data and files from one end to other. It needs to be a secure communication among the two parties [4].

This paper focuses on file transfer using Secure RSA, which eliminates some loopholes of RSA that might prevent a hacker from stealing and misuse of data. This paper also presents comparison between RSA file transfer and Secure RSA file transfer.

Keywords: file transmission, RSA algorithm, public key cryptography, private key cryptography

I. INTRODUCTION

In the current time, when the Internet provides essential communication between millions of people and is being increasingly used as a tool for ecommerce, security becomes a tremendously important issue to deal with. Internet is often used to upload web pages and other documents from a private development machine to public webhosting servers. Transfer of files like banking transactions e-shopping, tenders etc need special authenticated mechanism. As a communications and transmission of files over internet has increased exponentially since last few years, there is need of security in such file transfer. One of the solutions to secure communication is cryptography. It is the process of converting plain text into encrypted text and decrypt cipher text to plain text at other end.

In a distrusted medium cryptography becomes essential part of secure communication. There are two types of cryptographic algorithm to accomplish these goals: symmetric cryptography, asymmetric cryptography. The initial unencrypted data is referred as normal text.

Manuscript received on February, 2013.
Rajan.S.Jamgekar, Asst.Professor, NBNSCOE, Solapur, India.
Geeta Shantanu Joshi, Asst.Professor, MMCOEP, India.

It is encrypted into cipher text with a cryptographic algorithm, which will in turn be decrypted into usable plaintext. In symmetric cryptography single key is used for encryption and decryption e.g. Data Encryption Standard (DES) and Advanced Encryption Standards (AES).In asymmetric algorithm different keys are used to encrypt and decrypt the data.RSA is widely used in electronic ecommerce protocols. With sufficiently long keys and the use of up-to-date implementations; RSA is believed to be totally secure.

There are two ways in which we can achieve security 1.encrypted file transfer 2.Strong secure protocol for transmission of files.

RSA (Rivest, Shamir & Adleman) is asymmetric cryptographic algorithm developed in 1977. It generates two keys: public key for encryption and private key to decrypt message [2].RSA algorithm consist of three phases, phase one is key generation which is to be used as key to encrypt and decrypt data, second phase is encryption, where actual process of conversion of plaintext to cipher text is being carried out and third phase is decryption, where encrypted text is converted in to plain text at other side.

As a public key is used for encryption and is well known to everyone and with the help of public key, hacker can use brute force method to find private key which is used to decrypt message.

Secure RSA prevents files from hackers and help safe transmission of files from one end to other [2].

In this paper we introduce an algorithm that is a modification to the existing RSA algorithm. In our algorithm we have eliminated the need to send product of two random prime numbers in the public key. Further we have replaced the role of *n* in encryption and decryption by an integer.

This paper is organized as follows: In section 2, we give a brief review of existing asymmetric algorithms; RSA File Transmission is presented in section 3. In section 4, we have presented Implementation of RSA algorithm. Difference between RSA and MREA is presented in section 5; we give the conclusion in section 6.

II. RELATED WORK

Cryptography is a process which is associated with scrambling plaintext (ordinary text, or clear text) into cipher text (a process called encryption), then back again to plain text (known as decryption). The key feature of asymmetric cryptography system is encryption and decryption procedure are done with two different keys - public key and private key. Private Key cannot be derived with help of public key that provides much strength to security of cryptography.

This is one main difference between symmetric and asymmetric cryptography, but that difference makes whole process different.

This difference is small but it is enough that it has implications throughout the security. Mainly, symmetric cryptography is seen as faster, more lightweight, and better suited for applications that have a lot of data to transfer, while at the same time, it is known to be less secure and more open to wider areas of attacks because of maintenance of a private key required. This drawback is removed by asymmetric cryptographic algorithm discussed in following section.

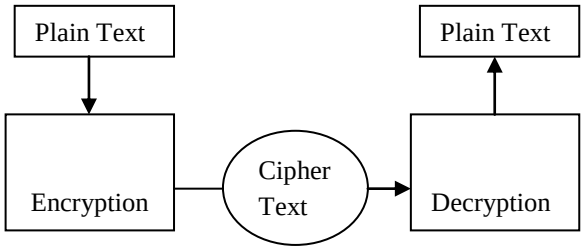


Figure 1. Symmetric Cryptography

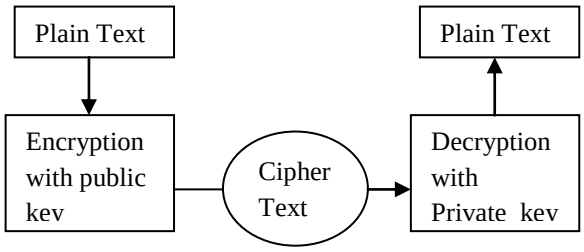


Figure 2. Asymmetric Cryptography

Elliptic Curve Cryptosystem (ECC)

Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz as an alternative mechanism for implementing public key cryptography. Elliptic curve cryptography (ECC) can provides the same level and type of security as RSA but with much shorter keys. Elliptic curve cryptography **(ECC)** is an approach of public-key cryptography based on the algebraic structure of elliptic curves over finite fields. Advantage of Elliptic curve cryptography is the public key and private keys have smaller size. The computation is fast as compared to other method and also it needs less storage space. Whereas the drawback of EC curves generation is complex, and difficult to implement a sustainable ECC algorithm [10]. However, implementers can rely on third parties for curves, which can be validated [13].

ElGamal system

The ElGamal system is a public-key cryptosystem based on the discrete logarithm problem [9]. It consists of both encryption and signature algorithms. The ElGamal signature algorithm is similar to the encryption algorithm in that the public key and private key have the same form; however, encryption is not the same as signature verification [10], nor is decryption the same as signature creation [10] .Signature creation depends on the ElGamal signature algorithm. The main disadvantage of ElGamal is the need for randomness, and its slower speed (especially for signing). Another potential disadvantage of the ElGamal system [10] is that message expansion by a factor of two takes place during encryption [11]. However, such message expansion is negligible if the cryptosystem is used only for exchange of secret keys ElGamal encryption is used in the free GNU Privacy Guard software [20], recent versions of PGP, and other cryptosystems [12]. ElGamal is not semantically secure.

Digital Signature Standard [14]

A digital signature is represented in a computer as a string of binary digits. A digital signature [14] is computed using a set of rules and a set of parameters such that the identity of the signatory and integrity of the data can be verified. An algorithm provides capability to generate and verify signatures. Signature generation [14] makes use of a private key to generate a digital signature. Signature verification [14] makes use of a public key which corresponds to, but is not the same as, the private key [14][15]. Each user possesses a private and public key pair. Public keys [14] are assumed to be known to the public in general. Private keys are never shared. Anyone can verify the signature [14] of a user by employing that user's public key. Signature generation [14] can be performed only by the possessor of the private key.

The advantages of this system are:

- The length of signature is shorter.
- The key generation is faster.
- The processing time cost is less.

Drawbacks of DSS are

- DSS and RSA are not compatible.
- The verification process is slower than RSA.

Diffie-Hellman key agreement protocol

Diffie-Hellman key exchange (D-H) is a cryptographic protocol that allows two parties that have no prior knowledge of each other to establish together a shared secret key over an insecure communications channel [1]. Then they use this key to encrypt subsequent communications using a symmetric-key cipher. The scheme [16] was first published publicly by Whitfield Diffie and Martin Hellman in 1976, Diffie-Hellman key agreement [17] itself is an anonymous (non-authenticated) key agreement protocol [1], it provides the basis for a variety of authenticated protocols [21], and is used to provide perfect forward secrecy in Transport Layer Security's short-lived modes as in. In the original description papers, the Diffie-Hellman exchange [17] by itself does not provide authentication of the communicating parties and is thus susceptible to a man-in-the-middle attack [14]. An attacking person in the middle may establish two different Diffie-Hellman key exchanges, with the two members of the party "A" and "B", appearing as "A" to "B", and vice versa, allowing the attacker to decrypt [23] (and read or store) then re-encrypt the messages passed between them. [23] A method to authenticate the communicating parties to each other is generally needed to prevent this type of attack [16]. The Diffie-Hellman algorithm depends for its difficulty of computing discrete logarithms. Secure Sockets Layer (SSL)/Transport Layer Security (TLS), Diffie-Hellman protocol is used in Secure Shell (SSH), Internet Protocol Security (IPSec), Public Key Infrastructure (PKI).

III. RSA FILE TRANSMISSION

RSA is widely used in encrypted connection, digital signatures and digital certificates core algorithms.

Public key algorithm invented in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman (RSA)[7]. It is the main operation of RSA to compute modular exponentiation. Since RSA is based on arithmetic modulo large numbers, it can be slow in constraining environments [18].

Especially, when RSA decrypts the cipher text and generates the signatures, more computation capacity and time will be required. Reducing modulus in modular exponentiation is a technique to speed up the RSA decryption. The security of RSA comes from integer factorization problem. RSA algorithm is relatively easy to understand and implement RSA algorithm is based on the theory of a special kind of reversible arithmetic for modular and exponent RSA is used in security protocols such as IPSEC/IKE, TLS/SSL, PGP, and many more applications [2][7]. The public and private keys are functions of a pair of large prime numbers and the necessary activities required to decrypt a message from cipher text to plaintext using a public key is comparable to factoring the product of two prime numbers.

RSA File Transmission Algorithm can be summarized as follows:

1. Generate the asymmetric keys with required digits.
2. Save and load the key, the key is saved as plain text.
3. Use specified key to encrypt any file with RSA algorithm.
4. Encrypted files can be loaded and decrypted with the specified key to restore the original file.

Secure RSA File Transmission

MREA is an asymmetric-key cryptosystem [20], meaning that for communication, two keys are required: a public key and a private key. Furthermore, unlike RSA [10], it is one-way, the public key is used only for encryption [10], and the private key is used only for decryption [10] [19]. Following is a key generation algorithm for MREA cryptosystem [19].

We have removed the drawback using Modified RSA (MERA)[19] for safe transmission of file from one user to other.

Secure RSA File Transmission Algorithm can be summarized as follows

1. Choose four large prime numbers p, q, r and s randomly and independently of each other. All primes should be of equivalent length.
2. Compute n = p x q, m= r x s, φ= (p-1) x (q-1) and λ=(r-1) x(s-1).
3. Choose an integer e, 1 < e < φ such that Gcd (e, φ) =1
4. Compute the secret exponent d, 1 < d < φ, such that e x d mod φ =1.
5. Select an integer g=m+1.
6. Compute the modular multiplicative inverse: $\mu=\lambda^{-1} \bmod m$.

The public (encryption) key is (n, m, g, e).

The private (decryption) key is (d, λ , μ)

Encryption:

Let F be a file to be encrypted where the contents of file are taken into string S.

Select random number r, where r < m.

Compute cipher text as: $c=g^{s\wedge emodn} \times r^{m} \bmod m^2$.

Decryption

Compute original message:

$S= (((c^{\lambda} \bmod m^2-1)/m) \times \mu \bmod m)^{d} \bmod n$.

IV. IMPLEMENTATION

The algorithm is implemented in JAVA. Difficulty of implementation mainly depends of the platform, applications and how much of the tools you need to implement from scratch. The algorithm successfully executes for .doc, .rtf, .txt, .java, all types of programmable files and other files having write permission. The class BigInteger is used to hold large prime numbers and keys so that it difficult for hacker to guess or use brute force method to find. Generation of random prime numbers gives the algorithm extra strength and efficiency. Euclid's algorithm is used to find GCD.

Modified RSA for secure file transmission algorithm is divided in to four parts

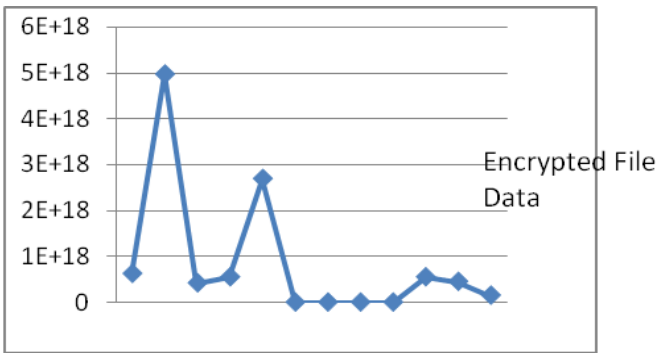
1. Selecting file for transmission
2. Encryption of file
3. Transmission of encrypted file
4. Decryption of file at other end.

32 bit keys are used to generate prime numbers and keys. The algorithm is tested and executed on 2.20 GHz Dual Core processor and one GB RAM. The algorithm is executed successfully on different types and different size of files. The summary of result is as follows

private key : 624256720618358291
ENCRYPT
ciphertext : 4961401011823809397
ciphertext : 419430400000000000
ciphertext : 550329031716248441
ciphertext : 2688592716557197975
ciphertext : 1792160394037
ciphertext : 100000000000
ciphertext : 31381059609
ciphertext : 31381059609

V. RSA AND MREA ALGORITHM

The algorithms (RSA & MREA) have many important parameters affecting its level of security and speed. By increasing the modulus length it is caused of increasing the complexity of decomposing it into its factors. This also increases the length of private key and hence difficulty to detect the key. Another parameter is modular multiplicative inverse μ where the modular multiplicative inverse μ is new factor of private key, so it will be more difficult to choose μ by trying all possible private keys (brute force attack) hence the security also increases as well as difficulty of detecting the private key. The RSA and MREA parameters are changed one parameter at a time and the others are kept fixed to study the relative importance. The results vary depending on type of file and size of file.



VI. CONCLUSION

MREA algorithm is used to encrypt files and transmit encrypted files to other end where it is decrypted. The project works efficiently for small size while it consumes time for large size of files. At a instant only one file can be encrypted and transmitted. As a future work multiple file encryption and decryption can be possible. It has broad development prospects. The project application was designed to take the efficiency and reusability into account. Great level of security is achieved using this algorithm. Modified RSA algorithm for file transmission algorithm can be used where high security file transmission required in public forums.

REFERENCES

1. Nan Li, "Research on Diffie – Hellman Key Exchange Protocol", IEEE 2nd International Conference on Computer Engineering and Technology, 2010, Volume 4, pp 634 – 637
2. Xin Zhou, Xiaofei Tang, "Research and Implementation of RSA Algorithm for Encryption and Decryption", IEEE, 6th International Forum on Strategic Technology, pp- 1118 – 1121
3. Eun- Jun Yoon, Kee –Young Yoo, "An Efficient Diffie – Hellman – MAC Key Exchange Scheme" IEEE, Fourth International Conference on Innovative Computing , Information and Control , pp 398 – 400, 2009.
4. Xi aowen Kang, Yingjie Yang, Xin Du,"A Disaster – Oriented Strong Secure File System", IEEE , 3rd International Conference on Innovative Computing Information and Control, 2008.
5. R . L. Rivest, A. Shamir and L. Adleman, "On Digital Signatures and Public Key Cryptosystems", Technical Memo 82, Laboratory for Computer Science, Massachusetts Institute of Technology, April 1970
6. Sonal Sharma, Saroj Hiranwal, Prashant Sharma,"A NEW VARIANT OF SUBSET-SUM CRYPTOSYSTEM OVER RSA",International Journal of Advances in Engineering & Technology, Jan 2012.ISSN: 2231-1963
7. R.L. Rivest, A. Shamir and L. Adleman, "A Method of obtaining Digital Signatures and Public Key Cryptosystems", Communication of the ACM, 21, 2(1978), pp 120-126
8. Sattar J Aboud, "An efficient method for attacking RSA scheme", IEEE 2009.
9. "A public key cryptosystem and a signature scheme based on discrete locarithms" TaherElGamal 1998,Springer-Verlag.
10. http://www.rsa.com/rsalabs/node.asp?id=2255
11. http://x5.net/faqs/crypto/q29.html
12. http://www.princeton.edu/~achaney/tmve/wiki100k/docs/ ElGamal_encryption.html
13. "Elliptic Curve Cryptography" Burt Kaliski.
14. "DIGITAL SIGNATURE STANDARD (DSS)", Federal Information Processing Standards Publication 186-2, 2000 January 27.
15. "DIGITAL SIGNATURE STANDARD (DSS)", Federal InformationProcessing Standards Publication 186, 1994 May 19
16. "DECISION SUPPORT USING MULTI SERVER AUTHENTICATION",BHAVNA CHANDRAN
17. http://simple.wikipedia.org/wiki/Diffie-Hellman_key_exchange
18. "The Research of the Batch RSA Decryption Performance", Qing LIU, Yunfei LI,Tong LI, Lin HAO,Journal of Computational Information Systems 7:3 (2011) 948-955
19. https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Merkle-Hellman.html
20. http://www.princeton.edu/~achaney/tmve/wiki100k/docs/ ElGamal_encryption .html
21. https://docs.fedoraproject.org/enUS/Fedora//html/Security_ Guide/apas02.html
22. http://en.wikipedia.org/wiki/Merkle%E2%80%93Hellman_ knapsack_cryptosystem
23. RFC 2631 – *Diffie–Hellman Key Agreement Method* E. Rescorla June 1999.

AUTHORS PROFILE

Mr. Rajan S Jamgekar, received Master of Engineering in Computer Science and Engineering with specialization in Computer engineering from Walchand College of Engineering, Sangali Maharastra-India He has 6 years of experience in teaching. His current research interest areas are Cryptography and Network. He has authored and co-authored more than 10 technical papers published in various prestigious national/international journals and referred conference, symposium, workshop proceedings.

Mrs. Geeta Shantanu Joshi, received Master of Engineering in Computer Science and Information Technology with specialization in Computer engineering from VIT, Pune Maharastra-India She has 10 years of experience in teaching. Her current research interest areas are Cryptography and Information security. She has authored and co-authored more than 12 technical papers published in various prestigious national/international journals and referred conference, symposium, workshop proceedings.