International Journal of Engineering Sciences & Research Technology

(A Peer Reviewed Online Journal)
Impact Factor: 5.164

IJESRT

RESEARCHERID
THOMSON REUTERS

[Agarwal *et al.*, 10(7): July, 2021]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

TAMPER PROOF FILES USING SECURE RSA

Archana Agarwal[*1], Satyam Pandey[2], Sachin Sethi[3], Raghav Popli[4] & Rishabh Tyagi[5]
[*1]Associate Professor, C.S.E department Inderprastha Engineering College Sahibabad, U.P., India
[2,3,4&5]Student, B.Tech C.S.E Inderprastha Engineering College Sahibabad, U.P., India

DOI: https://doi.org/10.29121/ijesrt.v10.i7.2021.3

ABSTRACT

In this paper we have introduced a secure RSA algorithm to create tamper proof file. There are several cases where we need to make tamper proof file for example to store a person's identity such as aadhar card, pan card, or any other legal paper etc [4]. In this paper we used RSA-SHA256 algorithm to create file tamper proof. The RSA algorithm is based on asymmetric key cryptography also known as Public Key cryptography. Two keys are generated in RSA, One key used for encryption & other one which is only known to authenticate receivers can only decrypt messages. No other key can decrypt the message. The RSA algorithm is a well known public key cryptography algorithm and was one of the first great advances in public key cryptography [24]. Even though it is an efficient algorithm it is vulnerable to attackers. With the help of all brute attacks, hackers can obtain private key.Many advancements have been done to improve RSA like BATCH RSA, MultiPrime RSA, MultiPower RSA, Rebalanced RSA, RPrime RSA etc [24]. As the use of the internet is increasing exponentially, it is used for many applications like email, chatting, and transferring data. This paper focuses on tamper proof files using Secure RSA, which eliminates some loopholes of symmetric cryptography that might prevent a hacker from stealing and misuse of data.

KEYWORDS: tamper proof files, RSA algorithm, public key cryptography, private key cryptography.

1. INTRODUCTION

In today's world, where the Internet provides essential communication between millions of people and is being more and more used as a tool for ecommerce, security becomes an enormously vital issue to wear down the security of the files[24]. The Internet is often used to upload web pages and other documents from a personal development machine to public web hosting servers [24].

Storing files like aadhar , pan card, legal documents etc need special mechanisms. As the number of sensitive digital documents over the internet has increased exponentially since the last few years, there is a need for security in file storage. One of the solutions is to use asymmetric cryptography to secure files. It is the process that converts the plain text into encrypted text or digital signature and decrypt ciphertext into plain text at the opposite end.

In a distrusted medium cryptography becomes an essential part to secure sensitive files. There are two types of cryptographic algorithms to accomplish the goals: symmetric cryptography, asymmetric cryptography. The initial unencrypted data is stated as traditional text.

It is encrypted into cipher text with a cryptographic algorithm, which will turn the decrypted content into usable plaintext. In symmetric cryptography the single key is used for encryption and decryption e.g. Data Encryption Standards (DES) and Advanced Encryption Standards(AES) etc. In an asymmetric algorithm different keys are used to encrypt and decrypt the data. RSA is widely used in the ecommerce protocols. With sufficiently long keys and the use of up-to-date implementations; RSA is believed to be totally secure [24].

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

RSA is an asymmetric cryptographic algorithm developed in 1977. It generates two keys for its functioning: public key for encryption and private key to decrypt message [2].RSA algorithm consist of three phases, First phase is the key generation which is to be used in encryption and decryption of the data, second phase is encryption, where the actual process of conversion of plaintext into ciphertext is being carried out and third phase is decryption, where encrypted text is converted in to plain text at opposite side.

As a public key is used for encryption and is well known to everyone and with the help of public key, hackers can use brute force method to find private key which is used to decrypt a message. Secure RSA prevents files from hackers and helps save files from tamper [2].

This paper is organized as follows: In section 2, we have given a brief review of existing asymmetric algorithms; Securing File with RSA algorithm is presented in the section 3. In section 4, we have presented Implementation of the RSA algorithm to make file tamper proof.

2. RELATED WORKS

File Encryption and Decryption Using RSA

Cryptography is defined as the 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 systems is encryption and decryption procedures are done with two different keys - public key and private key [24]. Private Key cannot be derived with the help of a public key that provides much strength to security of cryptography. This is one main difference between symmetric and asymmetric cryptography, but that difference makes the whole process different. The difference is small but it is enough that it has many implications throughout the security. Largely, symmetric cryptography is seen as faster, lightweight, and better suited for applications that have a lot of data to transfer, while at the same time, it is found to be less secure and more open to wider areas of attacks because of maintenance of a private key required [24]. This drawback is removed by an asymmetric cryptographic algorithm discussed in the following section.

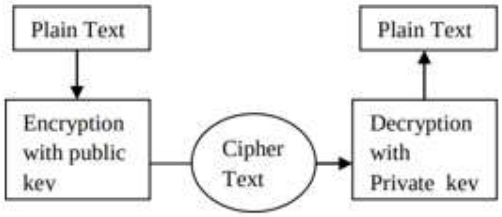

Figure 1. Symmetric Cryptography [24]

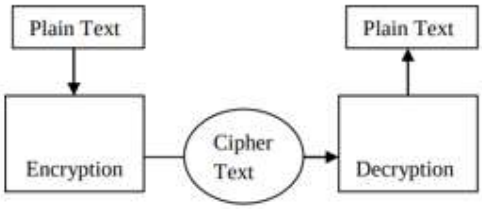

Figure 2. Asymmetric Cryptography [24]

Elliptic Curve Cryptosystem (ECC)

Elliptic Curve Cryptography (ECC) was invented in 1985 by Victor Miller & Neil Koblitz as an different mechanism for implementing public key cryptography. Elliptic curve cryptography (ECC) provides similar level and type of security as RSA algorithm but with shorter keys. Elliptic curve cryptography (ECC) is an approach

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

of public-key cryptography that supports the algebraical structure of elliptic curves over finite fields. Advantage of Elliptic curve cryptography is that the public key and private keys have smaller size. The computation is quick as compared to different strategies and additionally it needs less storage space. Whereas the drawback of EC curves generation is complicated, and troublesome to implement a sustainable ECC algorithm [10]. However, implementers will rely on third parties for curves that can be validated [13].

ElGamal system

The ElGamal system is a public-key cryptosystem supported on the discrete logarithm problem. It includes both encryption and signature algorithms. The ElGamal signature algorithm is comparable to the cryptography encryption algorithm wherein the public key and private key have the similar form, however, encryption is not the same as signature verification [10], nor is decryption the same as signature creation [10].

The main disadvantage of ElGamal is that the want randomness, and its slower speed (especially for signing). Another potential disadvantage of the ElGamal system [10] is that message growth by a factor of 2 takes place during encryption [11]. However, such message growth is negligible if the cryptosystem is used only for the exchange of private keys ElGamal encryption is employed within the free GNU Privacy Guard software [20], recent versions of PGP, and different cryptosystems [12]. ElGamal is not semantically secure.

Digital Signature Standard

A digital signature can be represented in a computer as a string of binary digits. A digital signature is computed employing a set of rules and a collection of parameters specifying the identity of the individual and integrity of the data to verify. An algorithm gives capability to create and to verify signatures. Signature generation makes use of a private key to create a digital signature for a file. Signature verification makes use of the public key but is not the same as the private key . Each and every user possesses a private and public key pair. Public keys are assumed to be familiar to the public in general. Private keys are never shared in public as they are important for the security of the files. Anybody can verify the signature of a user by making use of the user's public key. Signature generation can be done only by the possessor of the private key.

The advantages this method are:

- The length of the signature is shorter.
- The key generation is quicker.
- The processing time value is smaller.. Drawbacks of DSS
- DSS and RSA don't seem to be compatible.
- The verification method is slower than RSA.

3. SECURING FILES WITH RSA ALGORITHM

RSA is widely utilized in encrypted association, digital signatures and digital certificates core algorithms. Public key algorithm program was unreal in 1977 by Ron Rivest, Shamir Adi and Adleman Leonard . It is the main operation to cypher standard mathematical operation. Since RSA is predicated on arithmetic modulo giant numbers, it are often slow in confining environments [18]. Especially, once RSA decrypts the cipher text and generates the signatures, a lot of computation capability and time are going to be needed. Reducing modulus in standard mathematical operation could be a technique to hurry up the RSA decoding. The security comes from integer factorization problems. It is relatively easy to understand and implementation is based on the theory of special kind of reversible arithmetic for modular and exponent RSA is employed in security protocols such as IPSEC/IKE, TLS/SSL, PGP, and many other applications [2][7]. The general public and private keys are functions of pair of large prime numbers and necessary activities needed to decode a message from cipher text into plaintext using public key is similar to factoring the product of 2 prime numbers [24].

RSA Algorithm to make the file tamper proof can be summarized as follows:

1. Generate asymmetric keys with required digits.
2. Save & load the key, the key is saved as plain text.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

3. Use specific key to encode any file with the RSA algorithm.
4. Encoded files can be loaded and decoded with the specific key to verify the original file.

Tamper Proof Files
RSA-SHA256 algorithm creates digital signatures of files which are saved in a secure place, two keys required: public key & private key. Private key is used to generate digital signature for the sensitive files, since it's nearly not possible to come up with an equivalent digital signature employing a different private key, the digital signature for every file is unique and cannot be clone by another private key which provides tamper proof property. The public key can then be used to verify files using data from file and digital signature. Once we decode digital signatures into original files employing a public key we are able to then compare uploaded files with original files to check if somebody tries to tamper the files.

RSA Algorithm to make file tamper proof is to [24].

1. Choose four large prime numbers p, q, r and s indiscriminately and severally of every different. All primes ought to be of equivalent length.
2. Compute n = p x letter of the alphabet, m= r x s, φ = (p-1) x (q-1) and λ=(r-1) x (s-1).
3. Choose associate degree whole number e, 1 < e < φ such that
   GCD (e, φ) = 1
4. Compute the key exponent d, 1 < d < φ, such that (e x d) mod φ =1.
5. Select associate degree whole number g = m+1
6. Compute modular multiplicative inverse: $\mu = \lambda^{-1}$ mod m.

The public (encryption) key's (n, m, g, e) [24]. The private (decryption) key's (d, λ, μ) [24].

Encryption
Let F be a file to be encrypted where the contents of the files are taken into string S.
Select random number r such that r < m.

Decryption
Compute original file:
$S = (((g^{\lambda} \bmod n^2 - 1) / m) \times \mu \bmod m)^{n} \bmod n$ [24].

4. IMPLEMENTATION

The algorithm is implemented in Javascript. We used the keypair package in nodeJS to generate asymmetric keys and crypto package to generate digital signatures and to verify files. You can find implementation in the github link. The process can be encapsulated as follows:

1. First we generate a public and private key unique to the user, Let's say private key A and public key B.
2. Then we can use user private key A on hash value of file F using function G or RSA-SHA256 algorithm that can be written as: G(F, A)
3. G(F, A) gives us a digital signature of the file, that is unique for every file since no two private keys generate the same digital signature for the same file.
   i.e. G(F, A1) ≠ G(F, A2)
4. To verify files we can use public key B and digital signature G(F, A), now G(F, A) used as digest for verification algorithm V function i.e. V(G(F, A), B), now this function gives hash value H.
5. To check we can simply compare H with the hash value of uploaded file H', if H == H' then our file is safe but if H ≠ H' then someone tries to tamper .

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

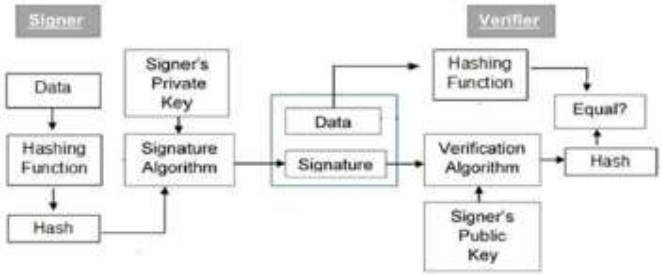

Figure 3. Model of Digital Signature [23]

Since signing large data through modular exponentiation is computationally difficult and time consuming. The hash of the data is a more or less small digest of the data, hence signing a hash is more organized than signing the entire data.

5. CONCLUSION

The RSA algorithm is used to encrypt files and to store it where it can be verified using public key to prevent tampering of files. The project works efficiently for small sizes while it consumes time for large files. Great level of security is achieved using this algorithm.

It has broad development prospects and can greatly improve security to store sensitive files. At an instant the system only can work on one file at a time. As a future work multiple file encryption and decryption can be possible.

6. ACKNOWLEDGEMENT

We feel honoured in expressing our profound sense of gratitude towards our mentor Archana Agarwal, Associate Professor, C.S.E. department, Inderprastha Engineering College, for providing us the opportunity to work on such a practical problem, under her benevolent guidance.

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
[4] –MAC Key Exchange Scheme” IEEE, Fourth International Conference on Innovative Computing , Information and Control , pp 398 – 400, 2009.
[5] 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.
[6] 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
[7] 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
[8] 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
[9] Sattar J Aboud, “An efficient method for attacking RSA schemes”, IEEE 2009.
[10] "A public key cryptosystem and a signature scheme based on discrete logarithms" TaherElGamal 1998,Springer-Verlag.
[11] http://www.rsa.com/rsalabs/node.asp?id=2255

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

[12] http://x5.net/faqs/crypto/q29.html
[13] http://www.princeton.edu/~achaney/tmve/wiki100k/docs/ElGamal_enc ryption.html
[14] "Elliptic Curve Cryptography" Burt Kaliski.
[15] "DIGITAL SIGNATURE STANDARD (DSS)", Federal Information Processing Standards Publication 186-2, 2000 January 27.
[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/MerkleHell man.html
[20] http://www.princeton.edu/~achaney/tmve/wiki100k/docs/ ElGamal_encryption .html
[21] https://docs.fedoraproject.org/enUS/Fedora//html/Security_Guide/apas 02.html
[22] http://en.wikipedia.org/wiki/Merkle%E2%80%93Hellman_knapsack_c ryptosystem
[23] RFC 2631 – Diffie–Hellman Key Agreement Method E. Rescorla June 1999
[24] https://www.tutorialspoint.com/cryptography/cryptography_digital_sig natures.html
[25] Mr. Rajan S Jamgekar, Mrs.Geeta Shantanu 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