Research article Available online www.ijsrr.org ISSN: 2279–0543

International Journal of Scientific Research and Reviews

Asymmetric Digital Signature Algorithm Based on Discrete Logarithm Concept with Execution Time Analysis to RSA

Arora Himanshu*, Shrivastava Sumit

*Research Scholar, Computer Science & Engineering,Sunrise University, Alwar (Rajasthan),India
Assoc. Prof., Computer Science & Engineering, Manipal University, Jaipur (Rajasthan), India

ABSTRACT

Cryptography is very important to continued growth of the Internet and electronic commerce. Digital signatures can also be used to authenticate the origin and the content of a message. RSA is the most popular asymmetric Digital Signature Algorithm. It uses a pair of keys, one of which is used to create the signature in such a way that it can only be signature verified with the other key. The keys are generated by a common process, but one cannot easily be generated from the knowledge of other. The security of the RSA system is based on the assumption that factoring of a large number is difficult. Due to advancement in technology and improvement in computation of speed it may become possible to break the Digital signature Algorithm like RSA, so a new technology is essential in times to come. An attempt in this direction is made in the this paper where a combination of factoring product of two large prime numbers as well as discrete logarithm problem is addressed simultaneously to improve the security of cryptosystem and examine the performance of two important digital signature algorithms.

KEYWORDS: Digital signature;Message digest; Random number, Discrete logarithm.

***Corresponding Author:**

Himanshu Arora

Research Scholar, Computer Science & Engineering,

Sunrise University, Alwar (Rajasthan), India

E Mail - arora_himansh@yahoo.com

INTRODUCTION

Public key cryptography is also referred to as asymmetric cryptography. It requires the use of both a private key (a key that is known only to its owner) and a public key (a key that is known to both of them)[1]. Public key cryptography is a widely used technology around the world. It is the approach which is employed by many cryptographic algorithms and cryptosystems. Public key cryptosystem offers both encryption and digital signatures[2]. Digital signatures are commonly used for software distribution, financial transactions, and in other critical security areas where it is important to safeguard against forgery and information tampering [3].

Concept of Digital Signature

Digital signatures are equivalent to traditional and written signatures in many respects. A digital signature (scheme) is a mathematical scheme for demonstrating the authenticity of a digital message or document. Digital signatures employ a type of asymmetric cryptography. For messages that are sent through a public channel, a properly implemented digital signature gives the receiver reason to believe that the message was sent by the claimed sender. In order to establish the authenticity of the electronic message, before a message is sent out, the sender of the message would sign it using a digital signature scheme (DSS) [3]. Thus, a digital signature scheme proves the authenticity of the message as well as the authenticity of the sender.

A digital signature scheme produce digital signature in a step by step fashion. For this, let us assume that the sender (A) wants to send a message M to the receiver (B) along with digital signature (DS) calculated over the message (M).

Step-1: The sender (A) uses the message digest algorithm to calculate the message digest (MD1) over the original message (M).

Step-2: The sender now encrypts the message digest with its private key. The output of this process is called as the digital signature (DS).

Step-3: Now the sender (A) sends the original message (M) along with the digital signature (DS) to the receiver (B).

Step-4: After the receiver (B) receives the original message (M) and the sender's (A) digital signature, receiver (B) uses the same message digest algorithm as was used by the sender (A) and calculate its own message digest (MD2).

Step-5: The receiver (B) now uses the sender's (A's) public key to decrypt (de-sign) the digital signature and output of this process is the original message digest as was calculated by sender (MD1) in step1.

Step-6: Receiver now compares both message digest MD1 (retrieved from sender's digital signature in step5) & MD2 (calculated in step 4), If both are identical then receiver can be quite sure that the original message and the digital signature came indeed from the sender.

Message Digest (also called a hash function) is a fingerprint or summary of a message [1, 2].

MATHEMATICAL BACKGROUND OF RSA

RSA (named after Rivest, Shamir and Adleman who first publicly described it) is an algorithm for asymmetric cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public key cryptography.

RSA is based on the principle that some mathematical operations are easier to do in one direction but the reverse is very difficult without some additional information. In case of RSA, the idea is that it is relatively easy to multiply but much more difficult to factor [4, 5]. Multiplication can be computed in polynomial time where as factoring time can grow exponentially proportional to the size of the number. Breaking of RSA algorithm is based on the technique that can factor the modulus n in a reasonable time [6,7]. Several different algorithms have been attempted to factor large numbers such as n. Fermat's Factorization and Pollard's algorithms are algorithms for finding a factor of n [8, 9]. The RSA algorithm works as follows.

1. Key Generation Process: First finds two prime numbers and generates a key pair using those two prime numbers.

   - p and q are distinct primes

     $n = p*q$

   - Find e, d such that : $e*d = 1 \bmod (p-1)(q-1)$

     Private Key: = (n, d)

     Public Key: = (n, e)

2. Message Digest Extraction Process: The message digest "m" is extracted from the original message "M" using MD5 algorithm.

3. Digital Signing and Verification Process: The signature creation and signature verification is done using the key pair.

   - Signature Creation : $S = m^d \bmod n$
   - Signature verification: $m = S^e \bmod n$

     (Where m = message digest)

Security of RSA algorithm is based on the mathematical fact that it is easy to find and multiply two large prime number together, but it is extremely difficult to factor the product and recover the two large prime numbers which have been multiplied earlier. It is, therefore, difficult for a hacker to identify the private keys from the knowledge of the public keys. This is known as factorization problem. Due to advancement in technology and improvement in computation speed it may become possible to break the RSA, so a new method is essential in times to come[10].

PROPOSED ALGORIHM & RELATED WORK

Proposed algorithm is similar to RSA algorithm with some enhancements. It uses an extremely large number having two prime factors (similar to RSA Digital Signature Algorithm). In addition to this, three natural numbers are also used. The improved method tends to increase the complexity of calculation. Security of Proposed algorithm depends on both factorization and discrete logarithm problem. Therefore, one has to solve both the problems to break proposed algorithm.

1. First generates randomly two prime numbers and three natural numbers which produce a key pair.

   - p and q are distinct primes and g, k and z are random numbers

     $N = p*q$

   - Find e, d such that : $e*d = 1 \bmod (p-1)(q-1)$
   - Find h, u such that : $h = gz$; $u = gk$

     Public Key: = (e, u, z, n)

     Private Key: = (d, h, k, n)

2. Then the Signature Creation and Signature verification is done using the key pair.

   - Signature creation = $(md \bmod n) \times hk. = s$
   - Signature verification = $(s/uz)e \bmod n = m$

If the above equation gets validated, then it can be said that the proposed algorithm really works.

Discrete Logarithm Problem

Given that elements g and h are in a finite group G, find an integer x such that

$$g^x \equiv h \pmod n$$

For example $3^x \equiv 13 \pmod{17}$ for x=4 is solution, but this is not the only solution since $3^{20} = 13 \pmod{17}$ is also a solution. This will also be applicable in case of $3^{4+16n} \equiv 13 \pmod{17}$ where n is an integer. Hence the equation has infinite solutions of form 4+16n. So discrete logarithmic approach is used, which is shown below:

$\log_e g^x = \log_e[h \pmod n]$

$x \log_e g = \log_e[h \pmod n]$

$x = \log_e [h \pmod n] / \log_e g$

$$x = \frac{\log_e[h(\bmod n)]}{\log_e g} * \frac{\log_g e}{\log_g e}$$

$x = \log_e [h \pmod n] * \log_g e$

where, $\log_e g * \log_g e = 1$

$x = \log_g [(h) \pmod n]$

In the equation given g, h and n, it is a straightforward method to calculate x. At the worst, one must perform repeated multiplications, so the discrete logarithmic concept is used to find out the value of x which was earlier used in the equation. The difficulty of discrete logarithmic problem is in the same order of magnitude as that of factoring prime numbers required for RSA.

SIMULATION RESULTS OF RSA AND PURPOSED ALGORITHMS

For the simulation purpose, purposed cryptosystem is implemented as a user-friendly GUI. This GUI application is implemented using JAVA library functions [12]. At Present one can enter either prime numbers (including random numbers) directly or can specify the bit length of the prime numbers & random numbers to be generated automatically [13]. Two prime numbers & three random numbers are involved in this implementation. Modulus is calculated with public and private keys of specific bit length which is generated using two prime numbers & three random numbers.

System Configuration:

Evaluation time is a machine dependent task which is required to be implemented on a particular system. Once the system configuration is changed, evaluation time will also be changed accordingly, however, in this work, following system configuration is used [11].

- Operating System: Windows XP Professional (5.1, Build 2600) Service Pack 2
- Processor: Intel Pentium Dual CPU E2200 @ 2.20GHz (2 CPUs)
- Memory: 1024MB RAM

In this simulation, Total execution time (T) is considered as a function of size of prime number, length of public key, chunk size and random number size ranging from 128 to1024 bits. One variable is kept constant while the other three variables are varied & results tabulated. Observation finally indicates, Total execution time (T) comprising of key generation time, digital signature creation time & digital signature verification time. Total execution time which is directly related to performance & security. The same is shown by the following tables and graphs:

Comparison basis on Changing the Bit Size of prime number p and q:

The bit size is concerned with the value of prime numbers p and q, as the given table and mentioned graph below, that when the value of prime numbers is increased, the overall execution time which comprises key generation time, digital signature creation time and digital signature verification time gets enhanced rapidly with the changes as depicted in Table 3.1 to3.4 & Figure 3.1 to 3.4.

TABLE 3.1 Bit size of prime no. v/s RSA, Purposed algorithm execution time when random number 16 bits.

Bit Size of prime number p and q	**Total Execution Time (ms) Purposed algorithm**	**Total Execution Time (ms) RSA**	**Difference in total execution time of Purposed algorithm and RSA (ms)**	**Percentage changes in execution time (%)**
128	174	169	5	2.958579882
256	610	594	16	2.693602694
512	3804	3732	72	1.92926045
1024	50118	49688	430	0.865400097

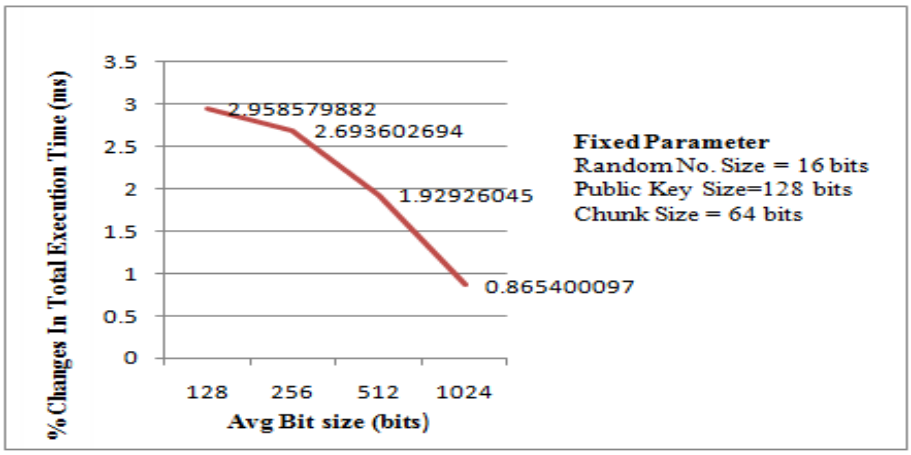

Fig.3.1 Bit size of prime no. v/s percentage changes in total execution time of RSA, Purposed algorithm when random number 16 bits.

TABLE 3.2 present Bit size of prime no. v/s RSA, Purposed algorithm execution time when random number 32 bits.

Bit Size of prime number p and q	Total Execution Time (ms) Purposed algorithm	Total Execution Time (ms) RSA	Difference in total execution time of Purposed algorithm and RSA (ms)	Percentage changes in execution time (%)
128	161	153	8	5.22875817
256	604	578	26	4.498269896
512	3689	3547	142	4.003383141
1024	25291	24844	447	1.799227178

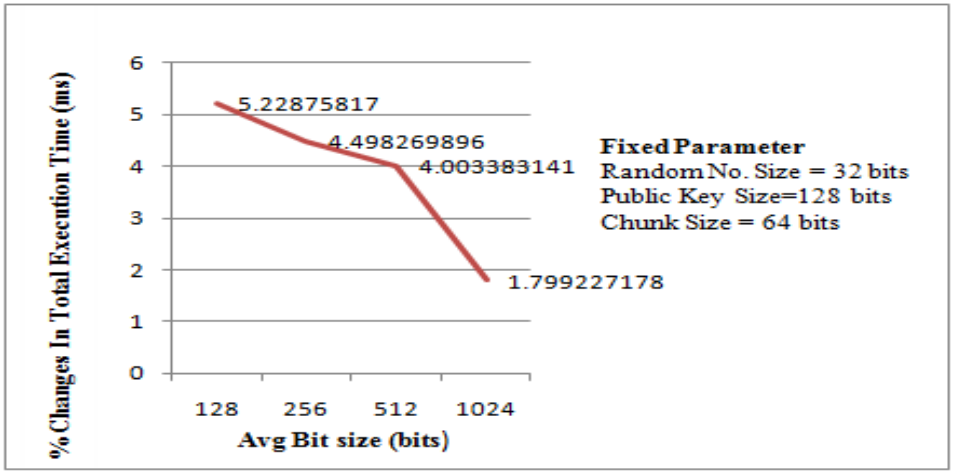

Fig.3.2 Bit size of prime no. v/s percentage changes in total execution time of RSA, Purposed algorithm when random number 32 bits.

Table 3.3 Bit size of prime no. v/s RSA, Purposed algorithm execution time when random number 64 bits.

Bit Size of prime number p and q	Total Execution Time (ms) Purposed algorithm	Total Execution Time (ms) RSA	Difference in total execution time of Purposed algorithm and RSA (ms)	Percentage changes in execution time (%)
128	203	188	15	7.978723
256	641	610	31	5.081967
512	3657	3500	157	4.485714
1024	25094	24609	485	1.970824

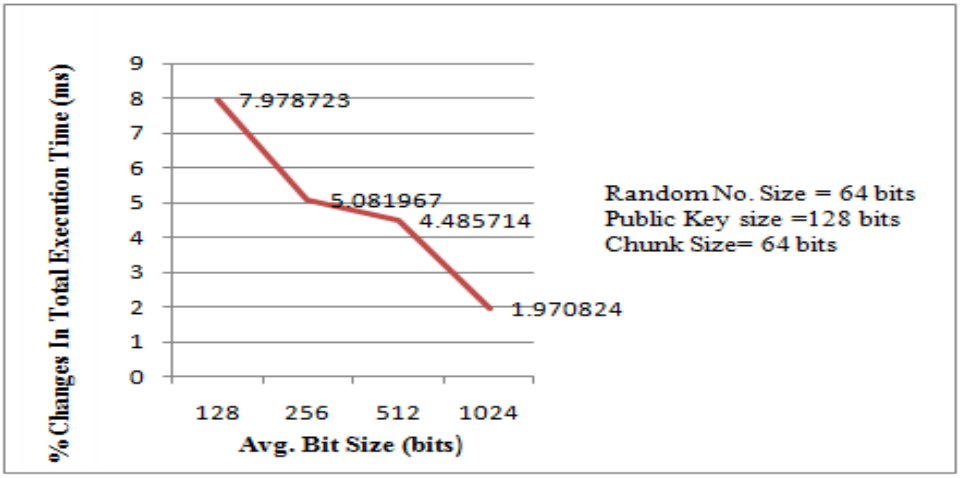

Fig.3.3 Bit size of prime no. v/s Percentage changes in total execution time of rsa, purposed algorithm when random number 64 bits.

TABLE 3.4 Bit size of prime no. v/sRSA, purposed algorithm execution time when random number 128 bits.

Bit Size of prime number p and q	Total Execution Time (ms) Purposed algorithm	Total Execution Time (ms) RSA	Difference in total execution time of Purposed algorithm and RSA (ms)	Percentage changes in execution time (%)
128	155	140	15	10.71429
256	598	562	36	6.405694
512	3552	3391	161	4.747862
1024	24281	23765	516	2.17126

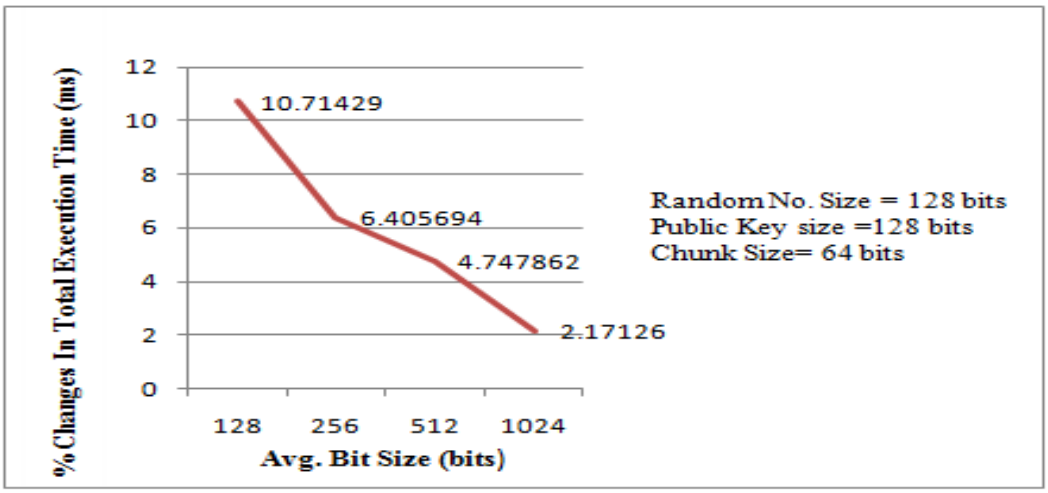

Fig.3.4 Bit size of prime no. v/s percentage changes in total execution time of RSA, purposed algorithm when random number 128 bits.

Here, we compare both the algorithms on the basis of Changing the Bit Size of prime number p and q. The size of prime number is increased from 128 to 256 the total execution time taken is less in proportion to when the bit size is increased from 512 to 1024 bits. This indicates that if enhanced security is to be achieved only on the basis of prime number then overall process i.e. signature creation & verification takes a very long time and the whole process becomes slow & tedious. As well as it is realized that when the average bit size is 128 bit then the percentage change in execution time is high in comparison to average bit size of 1024 bits which indicates that the impact of random number size is more on less average bit size.

CONCLUSION

The purposed algorithm extends the public key cryptosystem with its powerful digital signature technique utilizing random number generation concept. In order to enhance the security of algorithm in random, two improved ideas are proposed:

- Enhanced security is achieved by introducing random numbers.
- Establish more complex link between the random and the private key, so it is difficult for a hacker to use random number to indirectly attack the private key.

In the other aspects the order of time complexity purposed algorithm same as RSA schemes, the improved method only corresponds in the increase in complexity of calculation. All the calculation of random numbers is repeated each time. Considering purposed algorithm, the algorithm leads to more

complexity in breaking the digital signature. Security of purposed algorithm depends on both factorization and discrete logarithm problem. Therefore, one has to solve both the problems to break purposed algorithm.

REFERENCES

1 William Stallings, "Cryptography and Network Security Principles and Practices", ISBN-81-7758-774-9, Prentice Hall, Fourth Edition, 2006; 42-62,121-144,253-297.

2 Atul Kahate, "Cryptography and Network Security", ISBN-10:0-07-064823-9, Tata McGraw-Hill Publishing Company Limited, India, Second Edition, 2008; 38-41,59-73,87-122,153-196,205-240.

3 Hong Jingxin, "A New Forward-Secure Digital Signature Scheme", IEEE International Workshop on Anti-counterfeiting, Security, Identification, April 2007; 254-257.

4 R.Rivest, A. Shamir and L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, February 1978; 21 (2):120-126.

5 Bryan Poe, "Factoring the RSA Algorithm",Mat / CSC 494, April 27, 2005; 1-6.

6 Orhan K AKYILDIZ, "The RSA Cryptosystem", COMP5703 - Advanced Algorithms, October 21, 2008; 1-6.

7 AllamMousa,"Sensitivity of Changing the RSA Parameters on the Complexity and Performance of the Algorithm", ISSN 1607 – 8926, Journal of Applied Science, Asian Network for Scientific Information, 2005; 60-63.

8 Rajorshi Biswas, ShibdasBandyopadhyay, Anirban Banerjee, "A fast implementation of the RSA algorithm using the GNU MP library", IIIT – Calcutta, National workshop on cryptography, 2003; 1-15.

9 MykolaKarpinskyy , YaroslavKinakh, "Reliability of RSA Algorithm and its Computational Complexity" ,IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, September 2003; 8-10.

10 CHUK, Chinese university (2009), "RSA Algorithm security and Complexity", Retrieved from http://www.cse.cuhk.edu.hk/~phwl/mt/public/archives/old/ceg5010/rsa.pdf (26 Oct. 2009)

11 Li Xiao-fei. "An Improved ElGamal Digital Signature Algorithm Based on Adding a Random Number", Second International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC), April 2010; (2): 236-240.

12 Neal R. Wagner, "The Laws of Cryptography with Java Code", Technical Report, 2003; 78-112.

13 Wen-bi Rao, Quan Gan "The Performance Analysis of Two Digital Signature Schemes Based on Secure Charging Protocol", International Conference on Wireless Communications, Networking and Mobile Computing, Sept. 2005; 2: 1180 - 1182.