ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Maximizing Efficiency of RSA Cryptosystems by Improving Public Keys through True Randomization

Shaheen Hussain[1], Dr Nandita Sengupta[2]

Student, Bachelor of Science, Dept. of IT, University College of Bahrain, Kingdom of Bahrain[1]

Assistant Professor, Dept. of IT., University College of Bahrain, Kingdom of Bahrain[2]

ABSTRACT: RSA cryptography is one of the most commonly used methods to secure data transmission, through encryption of plaintext with the help of the public key. This public key is derived from two randomly selected prime numbers, which is long in length and complex, so that anyone who has the public key can encrypt a message. However, decryption of this message can only be done by knowing the private key, which includes the values of the two prime numbers and mathematical functions. Without the knowledge of these prime numbers, it is very difficult for the eavesdropper to decrypt the message. Considering the vulnerability of a man in the middle attack (MITM), it is necessary to take as many precautions as possible to encrypt the message without leaving an exposed approach the MITM could take to decrypt the message. True randomization of the parameters required for generation of keys is focused in our paper. Efficiency has been maximized through the true randomization process. This algorithm is focused for manufacturing industries, although can be applied to any organization that contains a database.

KEYWORDS: RSA algorithm; cryptography; true randomization; public key; Rabin-Miller primality test.

I. INTRODUCTION

The RSA cryptosystem was named after its designers, Ron Rivest, Adi Shamir, and Leonard Adleman in 1977. It is an asymmetric public key cryptosystem that is based on the complexity of factoring the product that is calculated from the two prime numbers, which is known as the factoring problem[1].

There are three phases in this public key encryption system. The first is key generation, in which the receiver of a secret message would create a public key, which is published, along with a private key that the receiver would keep secret. The algorithm to create these keys are complex, so that the MITM could not derive the private key from the public key if he were to retrieve the public key. The second phase is when the sender uses the receiver's public key to encrypt the secret message, which is then sent to the receiver. Once the message arrives to the destination, the receiver that already created the private key, would use this key to decrypt the message.

The RSA key generation process starts by selecting to prime numbers at random, which due to safety reasons should be between 512 to 1024 digits long. These prime numbers are labeled z and x. From these two numbers multiplied together the modulus is generated, called n. After this, the totient of n is calculated which results in φ(n), this can be determined by using the formula $\varphi(n) = \varphi(z)\varphi(x) = (z - 1)(x - 1)$. The value that is calculated is kept private. The reason why the RSA cryptosystem is vulnerable is due to the easy task it would be to generate the totient, only if the prime numbers have been determined. The next stage is to generate the public key.This is expressed by (e.n), which is generated by using another prime number, which is normally set at 65537. Using this default prime value might seemlike a security concern, however it is not because it is only being used for the public key, which will be shared. This is not a problem as long as the private key cannot be derived from the public key. The private key is generated by using the Extended Euclidean Algorithm. With respect to φ(n), this key is the multiplicative inverse of the public key.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

The common way to represent this decryption key is (d.n), which is created by using the following equation; $e \cdot d = 1 \bmod \phi(n)$.

The main purpose of true randomization to create the two prime numbers is due to the fact that there are many reasons RSA cryptosystems can be insecure. If the two prime numbers that are generated are small or close together, it would be easier to discover these numbers by the eavesdropper. The large prime number is generated by selecting a random large number, and then tested for primeness. This is commonly done by using the Rabin-Miller primality test, which has a high probability to determine if the selected number is prime. If it is not prime, then 1 is added to the number and tested again until two random prime numbers have been selected. My goal is to find the most secure way to generate the two prime numbers z and x.

II. RELATED WORK

In [2], the author, Burt Kaliski has worked with the creators of the RSA algorithm to explain the theory behind the mathematics that allows RSA to work. This is due to the fact that number theory is the foundation of computer security. He focuses on the process of generating the public and private keys from the two prime numbers being relatively easy, however finding the prime numbers from these keys is quite difficult.

Shown in [3], there are two ways to crack an RSA cryptosystem. They are mathematical attacks, and implementation attacks. In mathematical attacks, the goal is to factor n to find z and x, which would then allow the attacker to create a private key. The most popular implementation attack is called Kocher's Timing Attack. This is done when the attacker measures the time it takes for the decryption to take place, and processes this result to find the private key. The importance of the prime numbers being kept secret, or even destroyed after creation of the keys is essential, and shown in [4], which also recommends using the Rabin-Miller primality test in the process of checking for primeness.

In the book [5], the author mentions that to generate probable safe prime numbers, it is essential that they are over 512-bits. This leads to the reason why it is essential to create very large prime numbers randomly, that exceed the length of 512-bits, to increase the difficulty of the attacker factoring n to find them. In paper [6], authors mentioned random number generation options, their implementations in *gretl*. In paper [7], better metrics are explored for estimating entropy. It has also been mentioned that for random number generation there is an enormous gap between theory and practice. In book [8], Non-uniform random variate generation method has been discussed. Authors mentioned that researchers from various fields, like mathematics, statistics and computer science, are working in random number generation as it has wide applications and one of the applications is developing security algorithm. In paper [9], authors mentioned that inverse method of random number generation has many advantages with respect to other methods. In paper [10], it is mentioned that for security, randomness property plays an important role in AES.

III. PROPOSED ALGORITHM

A. *Design Considerations:*

- Hardware requirements must be low of cost.
- Computational time must be quick to find the two prime numbers.
- Attackers must not be able to find the random numbers.
- Random numbers must not be able to repeat themselves.

B. *Description of Proposed Algorithm for Random Number Generation:*

The main aim of this algorithm would be to use a complex unique number, multiplied with a dynamic value that would not repeat itself, to result in a prime number that is checked by the Rabin-Miller primality test. This algorithm contains seven processes grouped into four steps. Assume that this process is done by a manufacturing company.

Step 1: Finding two unique numbers:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Step 1 is consisting of four processes which will generate two numbers. The numbers which will be generated from step 1 will be multiplied by the numbers found from step 2 respectively. Result of these multiplications will be used as z and x respectively.

Process 1: A suitable way to find a number that is hard to recreate is by using biometric input. The user would first use a fingerprint scanner, and use this to scan any finger they wish to use. This would produce a series of 64-bit binary code, but to make it simpler, in this example a 32-bit code will be used, such as:

10110010
11010011
11010110
10110101

Process 2: In the 64-bit block, the process would be similar to process 1, however it is done by splitting the 64-bit block into two 32-bit blocks.

Process 3: This block of code would be used by the XOR function on the first 8 bits with the last 8 bits, then the middle two sets of 8 bits to create a new 16-bit binary code.

10110000
11010010

Process 4: These lines of binary digits would be concatenated to create the code “1011000011010010” which would equate to 45266 . Once this is done twice, with two different random fingerprints, we have two values that would be used to make up z and x.

45266
38573

Step 1 of proposed algorithm for random number generation has been represented in Fig. 1.

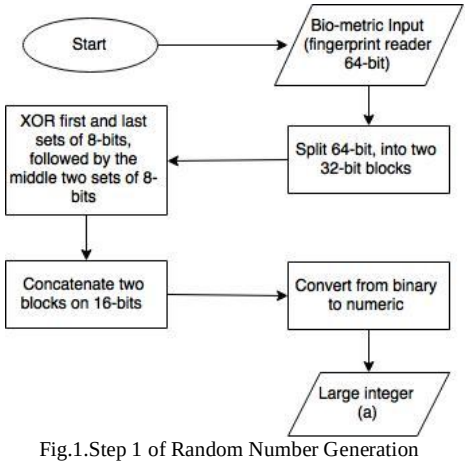

Fig.1.Step 1 of Random Number Generation

Step 2: Finding two dynamic numbers:

Process 5: Depending on the person that is generating the prime numbers, a random value in a database they are using, should be used to link a dynamic number with the generator. This is done by selecting a cell that contains a dynamic value, such as the amount in stock for a particular item, or group of items. This should be done with two different items or groups.

To demonstrate this, we can assume the person generating these random numbers is in a company that manufactures stationary equipment. For one item they would select the amount of red pens that are in stock, and another value of

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

their choice. Let us say that there is 76543 red pens in stock, and there are 4999 calculators in stock. Step 2 of proposed algorithm for random number generation has been represented in Fig. 2.

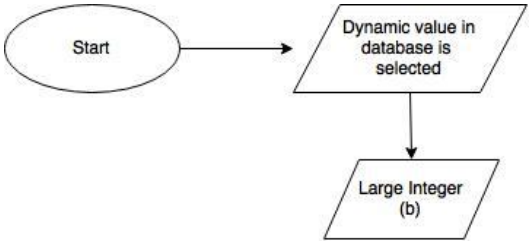

Fig. 2. Step 2 of Random Number Generation

Step 3: Creating the two prime numbers:

Process 6: Now that we have all four numbers, we can multiply them together.

45266 x 76543 = 3464795438

38573 x 4999 = 192826427

Step 3 of proposed algorithm for random number generation has been represented in Fig. 3.

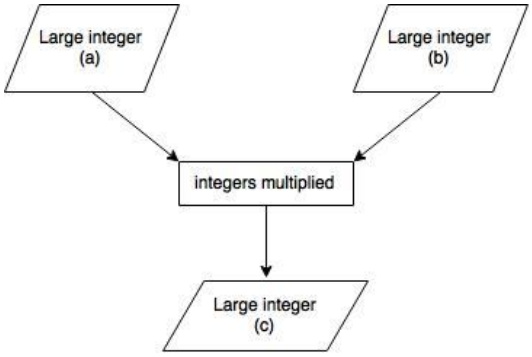

Fig.3. Step 3 of Random Number Generation

Step 4: Using the Rabin-Miller primality test:

Process 7: Now that we have two numbers we must test whether they are both prime, and if not we can add 1 to the number and retest it. The first value is not prime, so we have to add one and retest it until it is prime.

3464795438 = not prime
3464795439 = not prime
3464795440 = not prime
3464795441 = not prime
3464795442 = not prime
3464795443 = prime

192826427 = not prime
192826428 = not prime
192826429 = not prime
192826430 = not prime
192826431 = not prime
192826432 = not prime
192826433 = prime

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

z = 3464795443, x = 192826433.

Step 4 of proposed algorithm for random number generation has been represented in Fig. 4.

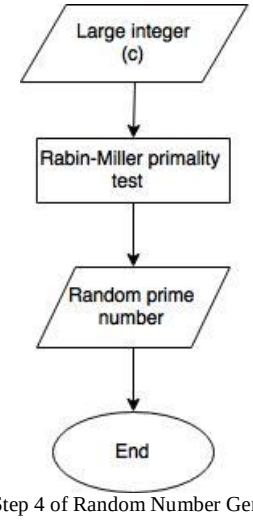

Fig.4. Step 4 of Random Number Generation

IV. PSEUDO CODE

Process 1: Retrieve biometric 64-bit binary block from fingerprint reader.
Process 2: Split the block into two halves of 32-bits each.
Process 3: XOR first and last sets of 8-bits, followed by the middle two sets of 8-bits.
Process 4: Concatenate the remaining 32-bits then convert it to its numerical value.
Process 5: Select a dynamic random value from a database, such as amount of a particular item stock.
Process 6: Multiply the values generated in process 4 and 5.
Process 7: Test for primeness with the Rabin-Miller primality test.
These processes are done twice to find the value for z and x.

V. SIMULATION RESULTS

To fully demonstrate the efficiency of this method with the RSA algorithm, the two prime numbers z and x from the proposed algorithm section will be used to generate an encryption key.

$$z = 3464795443, x = 192826433$$

The next step would be to find a value that will be the modulus for the public and private key, noted as n. This is achieved by multiplying z and x using eq(1).

$$n = z * x \qquad \text{eq(1)}$$

$$n = 3464795443 * 192826433$$
$$n = 668104146348344819$$

Once this is done, the totient ϕ(n) must be found from the prime numbers using eq(2).

$$\phi(n) = (z - 1) * (x - 1) \qquad \text{eq(2)}$$

$$\phi(n) = 3464795442 * 192826432$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

$\phi(n) = 668104142690722944$

Now that these values have been generated, all the required data is present to create the encryption and decryption key. The encryption key, which will be used as the public key will be noted as (e,n). The conditions for e are that it is relatively prime to ϕ(n).

$$e = 53$$
$$(e.n) = (53, 668104146348344819)$$

After the public key has been created, the private key (d,n) for decryption must be generated. This is done by using the following formula given in equation eq(3).

$$e * d = 1 \bmod \phi(n) \qquad \text{eq(3)}$$

In our proposed algorithm, calculation for finding the value of encryption and decryption key is very complex. Therefore, it will be difficult for the attackers to obtain the required value of d, for retrieving the confidential messages sent through the network.

VI. CONCLUSION AND FUTURE WORK

Due to the simulation results, it is clear that this method of random prime number generation works well with the RSA encryption algorithm. The proposed algorithm generates a very large encryption key which increases the difficulty of finding the prime numbers from the encryption key. Since a fingerprint reader isn't very costly, and the generation of the two prime numbers is fairly quick, this algorithm works well with the design considerations. On the other hand, due to the very large prime numbers, the computation time of the RSA algorithm would be long, which is the only disadvantage of this method, but worth the low cost with high security. Future work will be focused on using this method of randomization of biometrics and another value for other security purposes. There are many encryption methods that require number randomization and rather than using current techniques, such as sampling subatomic processes, would could be replaced with this method for a significantly lower cost. Uses of this proposed algorithm can be combined with the use of binary one-time pads, or used with the Elgamal Cryptosystem, which utilizes prime number randomization for its public key. To develop the complexity and randomness of this algorithm's products, a more expensive fingerprint reader can be purchased and used to generate a larger, more unique prime number.

REFERENCES

1. Boneh D., Stanford University, Ramarathnam Venkatesan, Microsoft Research, Breaking RSA May Be Easier Than Factoring.
2. Kaliski B., RSA Laboratories, The Mathematics of the RSA Public-Key Cryptosystem.
3. Cui L. X., Krishnan K., Attacks On the RSA Cryptosystem, 2005.
4. Digital Signature Standard (DSS), National Institute of Standards and Technology, June, 2009.
5. Mollin. A. R., 'RSA and Public-Key Cryptography', November, 2002.
6. Yalta T. A., Schreiber S., 'Random Number Generation in gretl', Journal of Statistical Software, Volume 50,August 2012.
7. Viega J., 'Practical Random Number Generation in Software', Proceedings of the 19th Annual Computer Security Applications Conference, 2003.
8. Hormann, W., Leydold, J., and Derflinger, G.,'Automatic NonuniformRandom Variate Generation', Springer-Verlag, Berlin, 2004.
9. Hormann, W. and Leydold, J., ' Continuous random variate generationby fast numerical inversion', ACM Transactions on Modeling and ComputerSimulation, 13(4):347–362, 2003.
10. Hellekalek, P. and Wegenkittl, S.,'Empirical evidence concerning AES', ACM Transactions on Modeling and Computer Simulation, 13(4):322–333, 2003.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

BIOGRAPHY

Shaheen Mohamed Hussain is a second year student at the University College of Bahrain, in the Department of Information Technology, completing a Bachelor of Science in Information Technology, with a concentration in Computer Science. His research interests are focused around computer and network security, with a high appeal to encryption techniques.

Dr. Nandita Sengupta has done her Bachelor of Engineering from IIEST formerly known as Bengal Engineering College, Shibpur, Calcutta University. She completed Post Graduate Course of Management in Information Technology from IMT. Later on she passed M Tech (Information Technology) from IIEST. She completed her PhD in Engineering (Computer Science and Technology) from IIEST. She has 25 years of working experience. 11 years she dedicated in industry. Last 14 years she is in academics and taught various subjects of IT. Presently she is working as Assistant Professor in University College of Bahrain, Bahrain. Her area of interest is Analysis of Algorithm, Theory of Computation, Soft Computing Techniques, Network Computing. She achieved "Amity Best Young Faulty Award" on the occasion of 9th International Business Horizon INBUSH 2007 by Amity International Business School, Noida in February,2007. She has around 25 publications in National and International conference and journals.