USING CHAOTIC MAPS TO ENHANCE RSA PUBLIC KEY CRYPTOGRAPHY

Ahmed T. Sadiq, E.M.E.Mostafa[1] ,Yasser F. Mahmoud[1] Ahmed B. Majeed[1]
Department of Computer Science, University of Technology, Iraq
[1]Department of Mathematics & computer science**,** Faculty of Science, University of Alexandria,Egypt
ahmed t. sadiq.naji@scbaghdad.edu.iq

ABSTRACT: *Complexity, confusion and diffusion are very important factors in cryptosystems, therefore there are several modification on cryptosystems to increase the performance of these factors. Asymmetric cryptosystems utilize 2 different keys, one of them is utilized for the encryption process and the other one is utilized for the decryption process, RSA crypto-system is based on the Integer Factorization Problem (IFP), in cryptography field, much attention is being given to Chaos. It does describe a system that is considered sensitive to the initial condition, The Chaos theory is utilized to generate a random number. In our research we utilize these issues to suggest a cryptosystem that is based on the IFP with chaos theory for the purpose of increasing the security of these techniques. Also, we are able to suggest a cipher-system which utilizes the chaos theory with the asymmetric algorithm in one system. For the purpose of increasing the security and complexity the Chaos function is applied, and for the purpose of generating random behavior, yet still entirely deterministic a chaos system is applied. In this study, we introduce two modifications regarding the RSA cryptosystem protocol depend on the IFP and chaos theory (1D & 2D Logistic Map) to keys generator as a security key and parameters used to more complexity against attacks.*

Keywords: Logistic map, Public key, Chaotic Theory, RSA Public Key Cryptosystem.

INTRODUCTION

Various approaches for asymmetric encryption (known as public-key encryption as well) will be discussed in this study. Concerning asymmetric encryption, any entity A has a public-key and a private-key (e, d) that corresponds to it. In secured systems, it's computationally impracticable to compute d when e is given. An encryption process Ee is defined by the public-key, while a related decryption transformation Dd is specified by the private-key. Each message m is being sent from B to A requires that B extract an authentic copy of e (public-key of A), ciphertext c = Ee(m) is obtained by utilizing the encryption transformation, then transmitting c to A. For the purpose of decrypting c, entity A utilizes the decryption process so as to get the original message m = Dd (c) [1]. Two keys are used in the public-key systems, one key (public-key) is utilized to encrypt the message and the other (private-key) is utilized to decrypt the message. The decryption system and the public key of the user could be revealed; however, it will still be very difficult to decrypt the messages (that is it does take a lot of computing during many years with utilizing the fastest computers in the world to compute it) [2]. A new perspective on cryptography field was introduced by the Public-Key Cryptography, Keeping the encryption technique secret is no longer the major idea behind encrypting data. In contrast, the receiving users in Public-Key Cryptography publish the technique used for data encryption along with their public keys [3].

In this paper an enhancement steps were proposed to increase the performance of RSA cryptosystem using chaotic maps. The next section present the related works, section 3 present the concepts of RSA cryptosystem. Section 4 illustrate the chaotic maps. Section 5 present the proposed enhanced RSA using chaotic maps. Section 6 illustrate the experimental results and evaluation. Finally, the conclusion in section 7.

1. Related Works

In 2017, Ahmed T. Sadiq et. al. proposed a public key encryption algorithm based on hybrid chaotic maps is proposed. The proposed algorithm uses a mixing of three dimensional Logistic map, three dimensional Arnold Cat map, two dimensional Rotation Equation and Chebyshev map to set random values to this algorithm and to generate privet and public keys that are used to encrypt and decrypt data. The experimental results show that the generated keys have the characteristics of truly random numbers and pass most of statistical and NIST tests [4].

In 2016, Hakem A and Alaa k. proposed a random binary sequence generator which generates a bit's sequence. The overall framework of the suggested model includes 2 parts, the first one is the non-deterministic source which is a mouse device, while the second one is a three-dimensional chaotic structure with the coordinates of the moving mouse's curser as the initial seeds and the resulted values are combined in the algorithmic procedure. The mouse cursor's coordinates are considered as an initial random number with post processing with three-dimensional chaos maps for the purpose of increasing the security and the randomness of the keys. The suggested study has a very long period and a high key space. In addition, the keys which were generated have effective statistical characteristics that is expected from genuently random binary sequence which are appropriate for using in critical cryptography systems, this is made via assessing the results through the hardness of sixteen tests of NIST which an abbreviation of (National Institute of Standards and Technology). [5]

In 2004, Gabor Vattay, Attila Fekete, Marjan Sterjev and Ljupco Kocarev proposed a Public key encryption with chaos, that it was a public key encryption algorithm on the basis of iterations of 1-D Chebyshev chaotic map and 2-D of torus automorphisms chaotic maps. The suggested encryption approaches are practical and secure, also it could be utilized in digital signature [6].

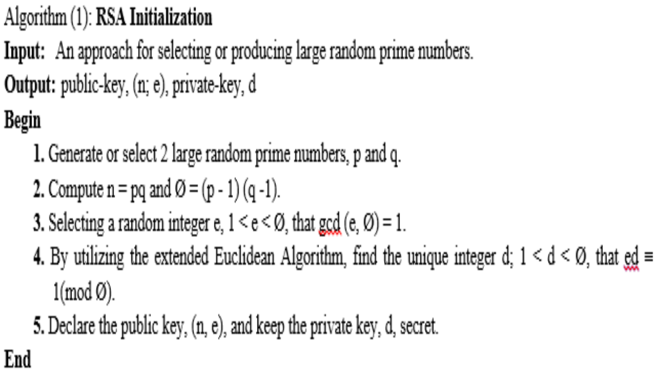
In 2007, Yoshifumi Nishio and Shuichi Aono proposed Cryptosystem on the basis of Chaotic map Iterations, a fresh cryptosystem through utilizing expansion chaotic map iterations. The expansion map is modified for the logistic map. The proposed cryptosystem is a symmetric-key cryptography that has characteristic of public-key cryptography that used 3 types of keys, a public-key, a private-key and a common private-key [7].

In 2009, R. Gnana Jayaraman, K. Ramar and K. Prasadh, suggested Public key crypto-systems on the basis of chaotic

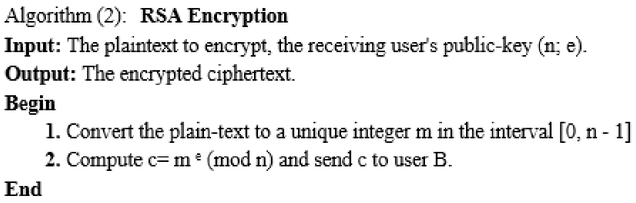
Chebyshev polynomials, an expansion of the public-key encryption that depends on Chebyshev polynomials with a modest hash function. The proposed model can be applied on multilevel inputs types such as images, and video [8].

RSA Public-key Encryption

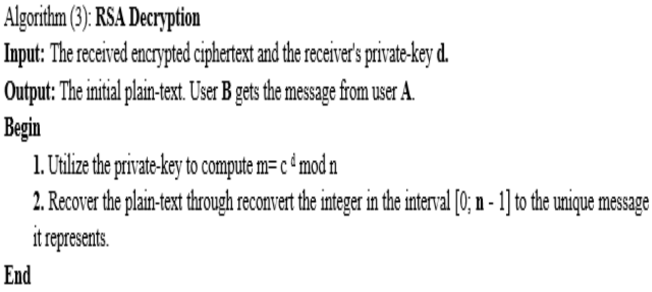
The RSA cryptosystem, named after its inventors R. Rivest, A. Shamir, and L. Adleman, is the most widely used public-key cryptosystem. It may be used to provide both secrecy and digital signatures and its security is based on the intractability of the fact that factorizing very large numbers are a `hard' problem. This problem is the Integer Factorization Problem discussed in Chapter two. [9]It was described to the public in August 1977, and was the invention of Ronald Rivest, Adi Shamir, and Leonard Adleman. However, it was recently revealed that this method was originally invented in 1973 within GCHQ by Clifford Cocks. Algorithms (1) through (3) describe the setup process required by each user upon initialization of the cryptosystem and the algorithms for encryption and decryption. In a group of users, each user must have access to each other user's public key, while retaining their own private key. When each person in the set of users has completed the initialization in Algorithms. [10,1]



```
Algorithm (1): RSA Initialization
Input: An approach for selecting or producing large random prime numbers.
Output: public-key, (n; e), private-key, d
Begin
    1. Generate or select 2 large random prime numbers, p and q.
    2. Compute n = pq and Ø = (p - 1) (q -1).
    3. Selecting a random integer e, 1 < e < Ø, that gcd (e, Ø) = 1.
    4. By utilizing the extended Euclidean Algorithm, find the unique integer d; 1 < d < Ø, that ed ≡
       1(mod Ø).
    5. Declare the public key, (n, e), and keep the private key, d, secret.
End
```



```
Algorithm (2): RSA Encryption
Input: The plaintext to encrypt, the receiving user's public-key (n; e).
Output: The encrypted ciphertext.
Begin
    1. Convert the plain-text to a unique integer m in the interval [0, n - 1]
    2. Compute c= m e (mod n) and send c to user B.
End
```



```
Algorithm (3): RSA Decryption
Input: The received encrypted ciphertext and the receiver's private-key d.
Output: The initial plain-text. User B gets the message from user A.
Begin
    1. Utilize the private-key to compute m= c d mod n
    2. Recover the plain-text through reconvert the integer in the interval [0; n - 1] to the unique message
       it represents.
End
```

Random Number Generator Using Chaos Theory

Chaos theory can be defined as a field related to mathematical science which examines the behavior of dynamic systems, as mentioned earlier, chaos theory is effective when utilized in designing random number generators [12]. Actually, chaotic system behaviors such as; high-sensitivity to initial states, mix up attribute, deterministic nature and also isn't capable of predicting the long term results [13], thus, those features are useful in cryptography field. For the purpose of generating large pseudo-random number, the logistic equation of chaos that is described below could be utilized. It is better to utilize many logistic equations during the generator design phase for the purpose of highly securing the keys and increasing the randomness of produced keys. An example of chaotic maps is logistic map (1D & 2D) as in the next equations:

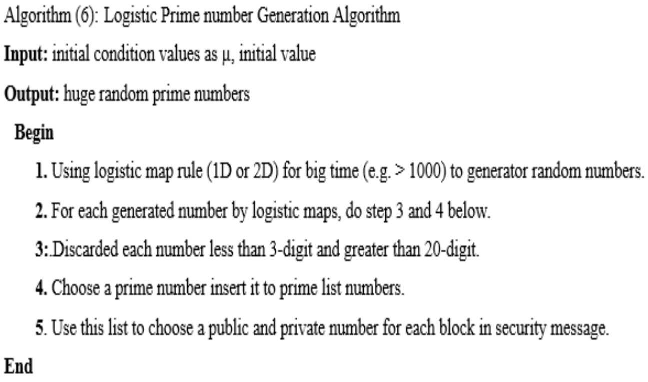
$$X_{n+1} = rX_n(1 - X_n)$$

Where X_n is a number between zero and one that represents the ratio of existing population to the maximum possible population. The values of interest for the parameter *r* (sometimes also denoted μ) are those in the interval [0, 4]. While the 2D logistic map equations are:

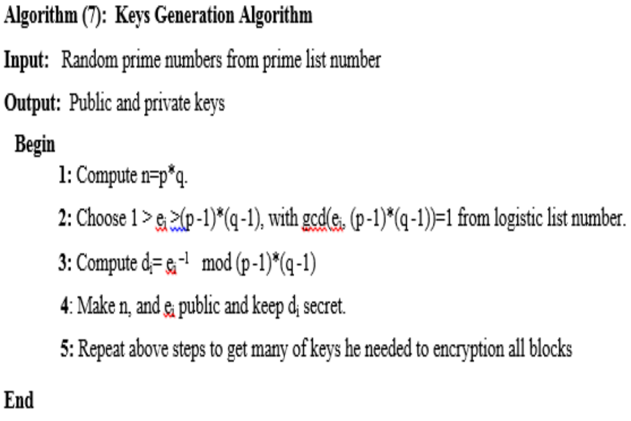
$$X_{n+1} = r(3Y_n + 1)X_n(1 - X_n)$$
$$Y_{n+1} = r(3X_{n+1} + 1)Y_n(1 - Y_n)$$

The Proposed Cryptosystems Depend on the IFP and Chaos Theory

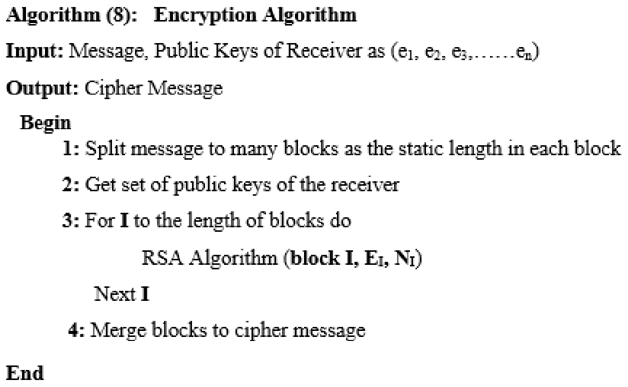
In this section, suggested cryptosystems based on logistic map and IFP in chaos theory will be introduced, the well-known technique in public-key cryptosystems which is an employee the IFP, is the RSA and its security depends on the intractability of factorization of quite large numbers are a "hard" problem. The second famous can use the 1D & 2D logistic map in chaos theory to generate huge numbers as random, after process these numbers and convert with select prime numbers in RSA parameters. The suggested technique is merging between random numbers and the RSA to make the suggested crypto-system better than the past original RSA. The proposal split the plain message to many bocks in fix size and each block cipher in special parameter depends on the key buffer with RSA techniques. This improvement increased the complexity of the system to make it more complex than the previous techniques.



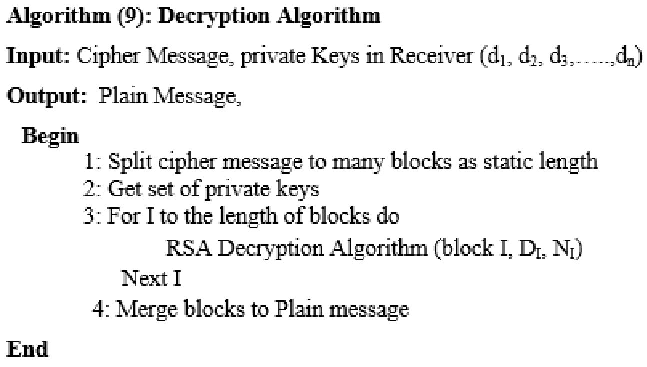
```
Algorithm (6): Logistic Prime number Generation Algorithm
Input: initial condition values as μ, initial value
Output: huge random prime numbers
 Begin
    1. Using logistic map rule (1D or 2D) for big time (e.g. > 1000) to generator random numbers.
    2. For each generated number by logistic maps, do step 3 and 4 below.
    3:.Discarded each number less than 3-digit and greater than 20-digit.
    4. Choose a prime number insert it to prime list numbers.
    5. Use this list to choose a public and private number for each block in security message.
End
```



```
Algorithm (7): Keys Generation Algorithm
Input: Random prime numbers from prime list number
Output: Public and private keys
 Begin
        1: Compute n=p*q.
        2: Choose 1 > $e_i$ ≥(p-1)*(q-1), with gcd($e_i$, (p-1)*(q-1))=1 from logistic list number.
        3: Compute $d_i$= $e_i^{-1}$ mod (p-1)*(q-1)
        4: Make n, and $e_i$ public and keep $d_i$ secret.
        5: Repeat above steps to get many of keys he needed to encryption all blocks
End
```



```
Algorithm (8):   Encryption Algorithm
Input: Message, Public Keys of Receiver as ($e_1$, $e_2$, $e_3$,……$e_n$)
Output: Cipher Message
  Begin
      1: Split message to many blocks as the static length in each block
      2: Get set of public keys of the receiver
      3: For I to the length of blocks do
              RSA Algorithm (block I, $E_I$, $N_I$)
          Next I
      4: Merge blocks to cipher message
End
```



```
Algorithm (9): Decryption Algorithm
Input: Cipher Message, private Keys in Receiver ($d_1$, $d_2$, $d_3$,…..,$d_n$)
Output:  Plain Message,
  Begin
      1: Split cipher message to many blocks as static length
      2: Get set of private keys
      3: For I to the length of blocks do
              RSA Decryption Algorithm (block I, $D_I$, $N_I$)
          Next I
      4: Merge blocks to Plain message
End
```

Example 1

In this example can explain how generated the random prime number using the logistic map in chaos theory depend on initial condition values and stage process ,this example is sample of numbers

1. Initialization

- μ=3.6 is perfect random values in logistic map
- Initial values X_0=0.99

2. Generation

In this step can generation many of real numbers depend on logistic equation 1-D in chaos theory when using the initial condition parameter:

0.03465,0.11707282125,0.36178371521098,0.808136905166921,0.542680766859531,0.868624232490988,0.399406613271504,0.83958339691272,0.471390907894264,0.872135319471099,0.3903035640075,0.832883421756902,0.487160196317958,0.874422988044923,0.38432599108229,0.828168332813169,0.498069408685187,0.874986954860113,0.382846742896588,0.826962900225217,0.500833416567102,0.87499756895889,0.382818881462229,0.826940049103829

3. Process

In this step must convert from real number to integer number by delete fraction part (0.) to get number can using as key public:

3465,11707282125,36178371521098,808136905166921,542680766859531,868624232490988,399406613271504,83958339691272,471390907894264,872135319471099,3903035640075,832883421756902,487160196317958,874422988044923,38432599108229,828168332813169,498069408685187,874986954860113,382846742896588,826962900225217,50083341656710287499756895889,382818881462229,826940049103829

4. Choose desired length

Delete small number loss than 3- digit and more than 20-digit (the length of number must all users used same length in filed can do it as secret protocol)

11707282125,36178371521098,808136905166921,542680766859531,868624232490988,39940,613271504,83958339691272,471390907894264,872135319471099,3903035640075,83288342176902,487160196317958,874422988044923,38432599108229,828168332813169,498069408685187874986954860113,382846742896588,826962900225217,500833416567102,87499756895889,382818881462229,826940049103829

5. Prime process

Choose only prime numbers ,this numbers can used as p, q as security parameter and from this list select also e as public key for each block or session between sender and receiver:

8136905166921,542680766859531,872135319471099,3903035640075,874422988044923,3832599108229,82123742973169,498069408685187,874986954860113,826962900225217,8749976895889,38281888146222,
826940049103829

Example 2

To Encryption and Decryption Message, the number used is small to calculate easy in example

1. Initialization

- Bob publicly generator a buffer logistic map depend on condition values.
- Bob selects 2 large prime numbers p_1 =1237 and q_1=4297 and choose another p, q for each block.

2. Key generation

In this algorithm must use the Euclidean algorithms to fast expositions and find the inverse number

- He compute n=p_1 * q_1= 1237 * 4297= 5315389
- Euler(n)=(p-1)*(q-1)=5309856.
- Choose 1 > e >= (p_1 -1)*(q_1 -1), with gcd(e_1, (p_1 -1)*(q_1 -1))=1,

Let e_1=8737,1 > 8737 >5309856,with gcd(8737, 5309856) =1.

- Compute d_1= e_1^{-1} mod (p_1 -1)*(q_1 -1),
  D_1= 8737^{-1} mod 5309856= 3714529.
- Make n_1, and e_1 public and keep d_1 secret.

3. Encryption

Alice sends the message M to Bob as follows, Alice split message to Blocks

- Let the first block is M_1=A=65.
- Depend on the public keys Bob used to encryption
- Compute cipher text C= $(M_1)^{e1}$ mod n_1,
- C = $(65)^{8737}$ mod 5315389 =4176547
- Transmit the C_1 is (4176547)

4. Decryption

Bob retrieves the message as follows

- Split the message to many of block depend on the secret protocol
- Let the first block is C_1=4176547
- Compute M = $(C)^{d}{}_1$=$(4176547)^{3714529}$ mod 5315389 =65**.**

Evaluation and Experimental Results

In this paper, the proposed work has been testing by the evaluation scales such as complexity, attacks, NIST and other tests.

Computational Complexity of the Proposed Method

In this part, the computational complexity of the encryption process and the decryption process of the RSA will be calculated, also generating the key in logistic map 1-D and 2-D chaos theory. The O-notation were very useful in supporting the analysts to categorize algorithms according to efficiency and in leading algorithm designer to search for the "best" algorithms for the significant problem. The main aim of studying the algorithm's computational complexity is to demonstrate that its running time is $O(f(N))$ for a certain function f. The Computational Complexity for RSA compared to the suggested techniques of encryption and decryption schema as bellow:

- **Original RSA Approach:**

The RSA's encryption scheme is:

$C = M^e \bmod n$ Then: $T(C)=O(\log n)^3$ bit operation.

The RSA's decryption scheme is:

$M = C^d \bmod n$ Then: $T(M)=O(\log n)^3$ bit operation.

The Proposed Method:

The encryption and decryption process of the suggested method 1 is: The proposal one used the same equation of RSA in encryption and decrypting so same complicity degree as:

$T(C)=O(\log n)^3$ bit operation.

But the different in calculate the secret and public parameter when in each session and in each block can used different keys as **P, Q, E,D** .so no method can find the complexity for the proposal or hard problem.

Key Space Analysis

Key space can be defined as one of the parameters utilized to measure the security of encryption algorithm. The security of the encryption improves when the key spaceis increased. The technique describes an encryption method which utilizes a text of other size. The key space is sufficiently large for resisting exhaustive attacks. The approach utilizes an input key size of more than 6 digits as (2^n) to breakable in brute force attacks. For the purpose of increasing the complexity, the initial condition in chaotic as X_0=0.99 and μ=3.6 in 1D and utilized another parameter in 2D logistic in range [1..4] for $\mu_{1,}$ μ_2 has to be symmetrical for generating same values and utilized in send and receive (i.e. encryption and decryption) .

Public Key Encryption based on Chaotic Maps

The proposed public key encryption based on chaotic maps is evaluated in this section. It is important to mention that the keystream generated from the chaotic maps is evaluated in this section. An efficient encryption algorithm has to withstand all kinds of attack. Numbers of test are used to evaluate the proposed algorithm and those tests are demonstrated in this section.

NIST Tests of Randomness

The randomness of the ciphertext generated after encrypting file is checked using **NIST** test of randomness as shown in following Tables. This result deepen on using 1D and 2D logistic map. From these results it is easy to see that the ciphertext passes most of NIST tests which means that it has good random properties.

Table (1) The Results of NIST Tests of the RSA with 1D and 2D Logistic Map Encrypted File

No.	Tests	1D	2D
1	Frequency	Success= 0.017010	Success= 0.017010
2	Block Frequency	Success= 0.024213	Success= 0.024213
3	Cumulative Sums	Success = 0.186156	Success = 0.186156
4	Runs	Success= 0.741618	Success= 0.741618
5	Longest Run	Success = 1.000000	Success = 1.000000
6	Rank	Success 0.000000	Success 0.000000
7	Discrete	Success= 0.096260	Success= 0.096260
8	Non-periodic Templates	Success =0.999252	Success =0.999252
9	Overlapping	Success = 1.000000	Success = 1.000000
10	Universal	Discard	Discard
11	Approximate Entropy	Success = 1.000000	Success = 1.000000
12	Random Excursions	Discard	Discard
13	Random Excursions	Discard	Discard
14	Serial	Failure =0.000000	Failure =0.000000
15	Lempel-Ziv Compression	Success = 1.000000	Success = 1.000000
16	Linear Complexity	Discard	Discard

Key Sensitivity

This test is used to check how the proposed algorithm resists against this kind of statistical attack. An efficient encryption method has to be sensitive for utilized secret key with the respect of initial parameters (in chaos system generator).A very small change in the initial parameter should produce a different output., from above tables it is easy to notice that a small change in the initial parameters in chaos a completely different for output.

Chosen Plaintext Attack

It is an attack model for cryptanalysis in which the attacker selects arbitrary plaintexts to be encrypted and obtains the corresponding ciphertexts. The proposed algorithms evaluated to check how it resists against chosen plaintext attack. From the key sensitivity attack results, it has been noticed that it is difficult for the attackers to produce the plaintext. Since, a small change in the initial parameters produce a completely different message. So, the proposed algorithm is robust against this kind of attack.

Known Plaintext Attack

In this attack, a part of plaintext is known by the attacker. In the proposed public algorithm, it is very difficult by the eavesdroppers to obtain correct private key from the value of initial conditions, parameter and public the key when comparing the plaintext with the ciphertext. So, the proposed system is effective against this kind of attack.

Brute Force Attack

The generated ciphertext from the three levels of security files are tested to check how they respondgains' the brute force attack. http://password-checker.online-domain-tools.com

Proposal Security

- Strength: 100% Approved.
- Evaluation:Chuck Norris approved.
- Dictionary Attack Check: Safe.

Brute-force Attack Cracking Time Estimate:

- Standard Desktop PC: About Infinity centillion years.
- Fast Desktop PC: About Infinity centillion years.
- GPU: About Infinity centillion years.
- Fast GPU: About Infinity centillion years.
- Parallel GPUs: About Infinity centillion years.
- Medium Size Botnet: About Infinity centillion years.

CONCLUSION

The most important different between the new proposal method and stander RSA (IF) public key cryptography using the prime numbers as secret keys or choose the secret parameter because those parameters are can change in each session for that is more difficult break from the attacker when we have buffer of keys are generator 1D and 2D from chaos system. The proposal against breakable depends on the logistic map and this equation is sensitive to the condition when a change in the initial parameter for this equation all output is different. The proposal method no need more timed because work in offline to generated and select all parameter for the public key algorithm.

REFERENCES

[1] M. P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.

[2] Rivest, Shamir and Adleman, “A method for obtaining digital signatures and public key cryptosystems", Communications of the ACM, Feb. 1978, pp. 120-126.

[3] T. ElGamal, “A public-key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Transactions on Information Theory, volume 31, pages 469-472, 1985.

[4] Ahmed T. Saadeq et. al., "A Proposed Public Key Encryption Based on Hybrid Chaotic Maps", the First International Conference on Information Technology (ICoIT'17), pp. 77-85, Erbil, Iraq, 2017.

[5] Alaa Kadhim F. and Hakeem Emad M.,” Mouse Movement with 3D Chaotic Logistic Maps to Generate Random Numbers”, Diyala journal of pure science, vol. 13, no. 3, July 2017, DOI: http://dx.doi.org/10.24237/djps.1303.268B , P-ISSN: 2222-8373 E-ISSN: 2518-9255.

[6] Shuichi Aono, Yoshifumi Nishio, "A Cryptosystem Based on Iterations of Chaotic Map", IEICE Technical Report, Vol.107, No.87, 2007.

[7] K. Prasadh, K. Ramar, R. Gnana Jayaraman, "Public-key cryptosystems based on chaotic Chebyshev polynomials", Journal of Engineering and Technology Research, Vol.1 (7), 122-128, 2009.

[8] Kamel Faraoun, "Chaos-Based Key Stream Generator Based on Multiple Maps Combinations and its Application to Images Encryption", The International Arab Journal of Information Technology, Vol. 7, No. 3, July 2010.

[9] M.O. Rabin, “Digitalized signatures and public-key functions as intractable as factorization”, MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[10] Monteserrat B.,”Hyperelliptic curve cryptosystem over optimal extension fields”, bachelor of engineering,thesis,univ. of Queensland,2000.

[11] W. Stallings,” Cryptography and Network Security, Principle and Practice”, Addison Wesley, 1999.

[12] M. François, T. Grosges, D. Barchiesi and R. Erra, “A New Pseudo-Random Number Generator Based On Two Chaotic Maps”. Informatica, Vol. 24, No. 2, pp. 181–197, 2013.

[13] J.M. Bahi, C. Guyeux, and Q.Wang. “A Pseudo-Random Numbers Generator Based On Chaotic Iterations. Application to watermarking”, International Conference on Web Information Systems and Mining, Vol. 6318 of LNCS, pp. 202–211, Sanya, China, October 2010.

[14] Strogatz and Steven, “Nonlinear Dynamics and Chaos", Perseus Publishing, 2000.