MULTI KEY ALGORITHM FOR SECURITY ENHANCEMENT OF RSA ALGORITHM USING MATLAB

Atyaf H. Muttaleb [1], Mohammed J. Zaiter [2]
[1,2] Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
atyaf954@gmail.com [1], mjzaiter@eetc.mtu.edu.iq [2]
Received:17/12/2019, Accepted:22/1/2020

***Abstract*- In this paper, a new technique is deployed to afford a strong secure algorithm based on dynamic keys strategy to enhance the security of network against cyber- attacks. The objective of this technique is to provide confidentiality and authentication of data between any partners. The proposed technique takes Multiple keys pair (Public and Private) with RSA algorithm which every key pair used to ensure the security of one communication session. Where the proposed technique enhances the communication system by defining a set of ciphers such that not only the key but also the cipher changes on every new data transaction. Simulation results have been achieved using MATLAB R2014a; Single Key RSA (SK- RSA) and proposed Multi Keys RSA (MK- RSA) are compared analysis is made on execution time and secrecy of the communication system. Hence, deploying the MK- RSA algorithm is proving more secure than SK- RSA algorithm with the advantage of no need time of execution for key generation in every communication sessions.**

I. INTRODUCTION

With the evolution of communication systems and information technology, data that is transmitted over networks have become more valuable. This leads to catch up the attention of attackers and researchers to focus in this area. Where researchers proposed several solutions based on cryptography methods. The classification of the cryptographic system is identified with three distinct dimensions used for the size of data manipulate, the number of the key used and the way of converting data [1]. With the number of keys, dimension cryptography is split into two categories as symmetric-key cryptography or privet- key encryption and asymmetric cryptography or public- key encryption [2]. The asymmetric cryptographic add many application as Encryption/ Decryption, Digital Signature and Key exchange. The most widely used algorithm in the asymmetric cryptographic is RSA (Rivest- Shamir- Adleman) which, proposed by Ron Rivest, Adi Shamir and Leonard Adleman [3]. The RSA have problem with require a big exponential multiplications modulo a large integer N of encryption and decryption in RSA. The authors of [4] present different methods to enhanced the RSA algorithm, which contributed an improvement of RSA in terms of the time of implementation. In [5] the authors proposed a variant of RSA to simplify the operation of decryption and digital signature at the receiver side.

II. TRADITIONAL RSA ALGORITHM

In RSA two keys are generated: for encryption public key and private key to decrypt the message. RSA algorithm consist of three steps, the first step is keys generation that is used to encrypt and decrypt data, the second step is encryption, perform actual process of transformation of the plaintext into ciphertext, and third step is decryption, where encrypted text is translated into plain text at the receiver [6], [7] as shown in Fig. 1 .

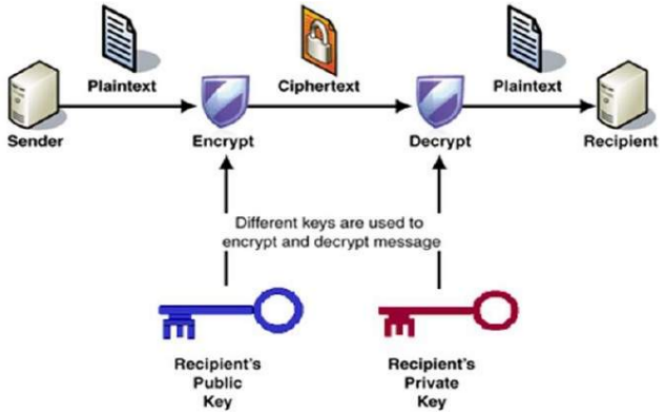

Figure 1: Asymmetric cryptography technique

A. Key generation

The Keys of RSA algorithm generated by multiply two large prime numbers. Where first select public key and then derived private key as the following Algorithm [8], [9].

Algorithm (1):

Select two prime numbers p and q where $p \neq q$

$n \leftarrow p * q$

$phi(n) \leftarrow (p-1)(q-1)$

Select the value of e where e $\in \{1, 2, ..., phi(n) - 1\}$

$and \gcd(e, phi(n)) == 1$

Calculate the value of d such that $d \cdot e \equiv 1 \, mod \, phi(n)$

Public key: PU = (e , n)

Private key: PR = (d , p , q)

B. Encryption method

If a sender wants to send the information to the receiver, the receiver should send the public key to the sender first, and then the sender will encrypt the message(x) with a public key (e). Hence, compute the ciphertext (y) by using the following equation [10], [11]

$$y = x^e \, mod(n) \tag{1}$$

C. Decryption method

Decryption will restore the encrypted text (y) to the original message (x) by using encrypted text (y) and private key (d). Then the original message is compared if it's equal to the decrypted message, in that case the algorithm is proved [12], [13]

$$x = y^d \, mod(n) \tag{2}$$

D. Digital signatures

Digital signatures have been done by reversing the operation of encryption and decryption method. Where first use the private key (d) of the sender to fix the digital signature in the message (M) as the equation below [14].

$$S = M^d \, mod(n) \tag{3}$$

At the receiver, received the digital signature (S) and verified by using the sender public key (e), as the following equation:

$$M = S^e \, mod(n) \tag{4}$$

E. RSA with fast exponentiation

Fast Exponentiation is used to hasten the processing of the RSA algorithm by minimizing the number of multiplication. Exponentiation, which will be utilized within both Encryption and Decryption. This algorithm decreases the computation time with the modular exponentiation complexity is also decreased. The equation $a^b mod\, n$ could be calculated faster by converting the value of b to the binary scheme, then do the square and multiplication. The detail of operation is cleared in the following Algorithm [15]:

Algorithm (2):

$c \leftarrow 0$

$f \leftarrow 1$

for $i \leftarrow k$ down to 0

do $c \leftarrow 2 * c$

$f \leftarrow (f * f) mod\, n$

if $b_i = 1$

then $c \leftarrow c + 1$

$f \leftarrow (f * a) mod n$

return f

Note that the variable c is not needed; it is included for explanatory purposes. The final value of c is the value of the exponent while the f represents the solution of the formula. Note: The integer b is expressed as a binary number $bk\, bk - 1 ... b0$.

III. PROPOSED MULTI KEYS RSA (MK-RSA)ALGORITHM

A. Key generation

In key generation process of MK- RSA, we will generate multiple public and private keys. Hence, generating multiple keys strategy to ensure the security of communication systems. Where public keys are visible to both sender and receiver while private keys are kept secret for the two partners. The key generation steps for user A as the proposed procedure are given as the following Algorithm :

Algorithm (3):

Select two prime numbers P and Q where$P \neq Q$

$N \leftarrow P * Q$

$H(N) \leftarrow (P-1)(Q-1)$

Calculate a grub of integer ei such that $H(N)/2 < ei < H(n) and \gcd(ei, H(N)) == 1$

Compute values of di such that $di \cdot ei \equiv 1 \, mod(H(N))$ Public Keys $P \cup a(i) = \{ei, N\}$,

Private keys $PRa(i) = \{di, P, Q\}$

While, the key generation steps for user B are given as following algorithm:

Select two prime numbers R and S where $R \neq S$

$Z \leftarrow R * S$

$K(Z) \leftarrow (R-1)(S-1)$

Calculate a grub of integer gj such that $K(N)/2 < gj < K(Z) \, and \, \gcd(gi, K(Z)) == 1$

Compute values of tj such that $tj \cdot gj \equiv 1 \, mod \, (K(Z))$

Public Keys$P \cup b(j) = gi, Z$,

Private keys $PRb(j) = \{tj, R, S\}$.

For more details the following flowchart illustrates the key generation steps of the proposed algorithm for user A. Hence, the algorithm selects big values of public exponent from $(H-1)$ to $(H/2)$ to avoid small public exponent attacks [16] shown in Fig. 2.

B. Confidentiality and authenticity

The RSA algorithm can be used both for message confidentiality and message authenticity. For achieving confidentiality and authenticity as shown in Fig. 3, where the, the sender (user A) encrypts a message (M) with the first private key $PRa(i) = \{d1, P, Q\}$ to achieve the authentication (digital signature of user A) $C1$ as illustrate in equation. 5

$$C1 = M^{d1} mod(N) \tag{5}$$

And then encrypts C1 by using receiver's (user B) first public key $P \cup b(1)\{g1, Z\}$ to achieve the confidentiality:

$$C = C1^{g1} \, mod(Z) \tag{6}$$

After two encryption process as explained above the ciphertext (C) send to the channel. At the destination B, the ciphertext (C) received and then decrypts by using the first receiver's private key $PRb(1)\{t1, R, S\}$ to verify the message confidentiality:

$$C1 = C^{t1} \, mod(Z) \tag{7}$$

And then verify the Digital Signature of user A (sender) by using sender's first public key $P \cup a(1) = \{e1, N\}$ to verify the originality of message from sender A:

$$M = C1^{e1} \, mod(N) \tag{8}$$

If there is another message from user A, to achieve confidentiality and authenticity, return does the same above steps, but with different keys exactly the second key in all steps. It is used for user A public key $P \cup a(2) = \{e2, N\}$, private key$PRa(2) = \{d2, P, Q\}$ and for user B public key Pub $(2) = \{g2, Z\}$, private key$PRb(2) = \{t2, R, S\}$. While if user B have a message to send to user A with message confidentiality and message authenticity. In this case, do the same above steps to achieve confidentiality and authenticity, but user B became the sender and user A became the receiver with the next keys in the list of sender and receiver to achieve the dynamic keys to ensure the security of communication systems.

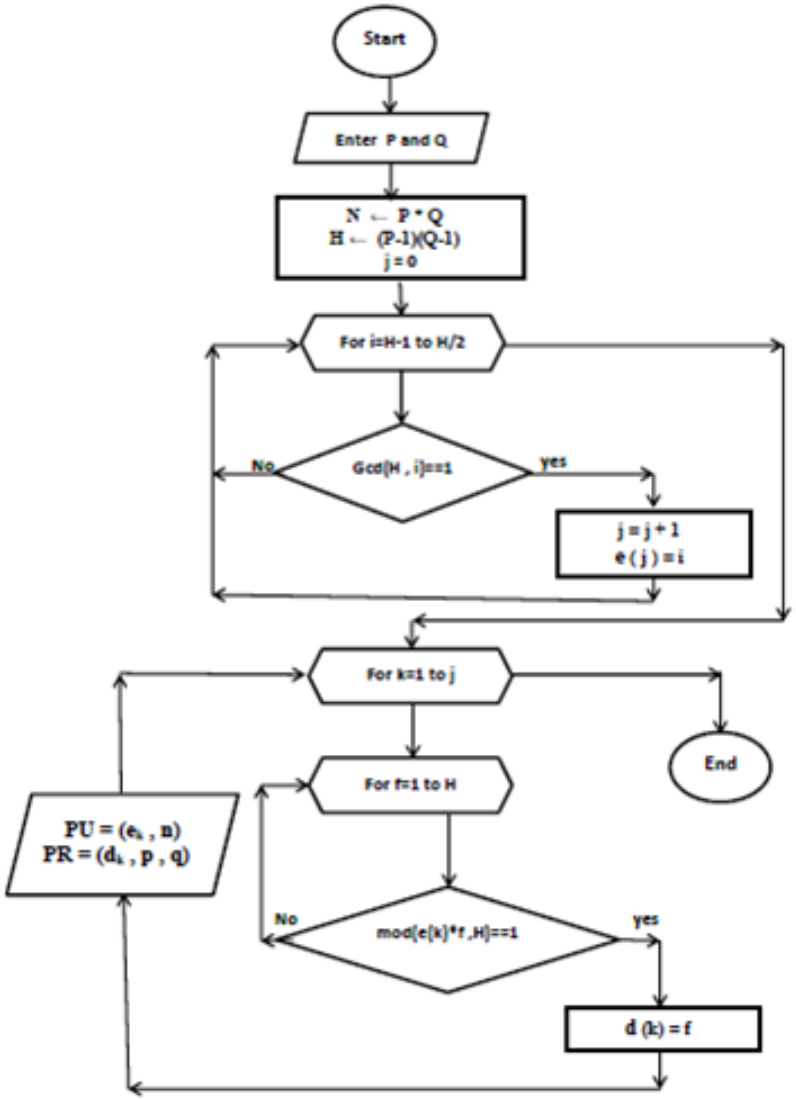

Figure 2: Flowchart showing the key generation steps of the proposed algorithm for user A

IV. IMPLEMENTATION AND RESULTS

A. *Single key RSA (SK- RSA) algorithm implemented for confidentiality and digital signature*

1) Run the RSA- SK.m Matlab code in Matlab software (here Matlab R2014a based simulation used).
2) It asks for entering the prime numbers values P and Q for user A, and R and S for user B as shown in Fig. 4 .
3) After entering values for user A and user B, the program start to generate public and private key then, keys generated for user A and B, where elapsed time for the generation is calculated.
4) Entering the send message at the source (user A) as in the Fig. 5
5) At the source converting the message letters' to its corresponding ASCI code as shown in Fig. 6.
6) The digital signature of the sender is computing according to Equation 5.
7) Encryption of the output of step 6 is computed according to Equation 6.
8) At the destination (user B) the received message first is decrypted according to Equation 7.
9) Verify the signature using equation 8 to achieve authentication to retrieve the original message, the elapsed time for the implementation steps 5 - 9 is calculated as shown in Fig. 6.

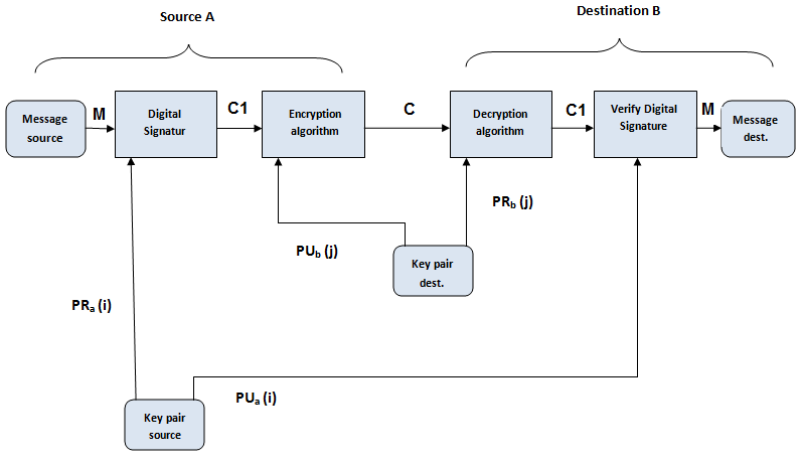

Figure 3: Confidentiality and digital signature for proposed multi keys RSA (MK- RSA) algorithm

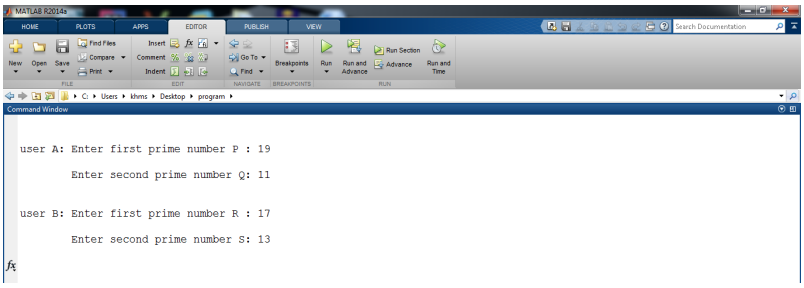

Figure 4: Implementation of SK- RSA algorithm for values of P, Q, R and S

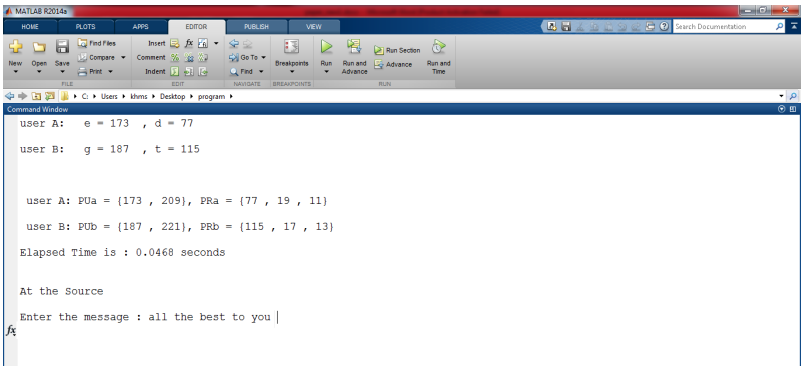

Figure 5: Key generation and entering the send message

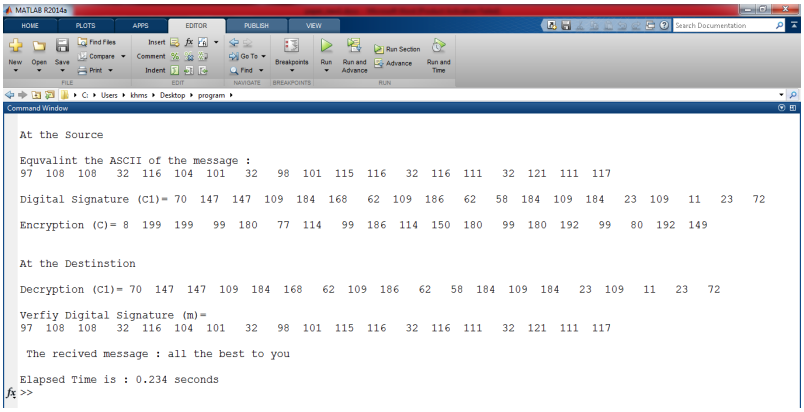

Figure 6: Computing digital signature and encryption at the source, and destination for SK- RSA

B. Proposed multi keys RSA (MK- RSA) algorithm implemented for confidentiality and digital signature

1) Run the RSA- MK.m Matlab code in Matlab software (here Matlab R2014a based simulation used).
2) It asks for entering the prime numbers values for P and Q for user A and R and S for user B as shown in Fig. 7.

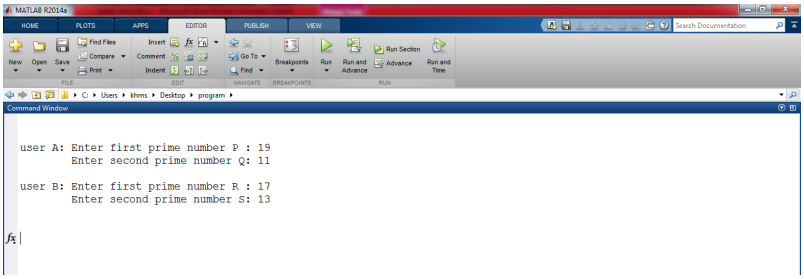

Figure 7: Implementation of MK- RSA algorithm for entering values of P, Q, R and S

3) After entering values for user A and user B, the program starts to generate multiple public and private keys which are 23 keys pair for user A and user B. Hence, the elapsed time for keys generation is calculated as shown in Fig. 8.

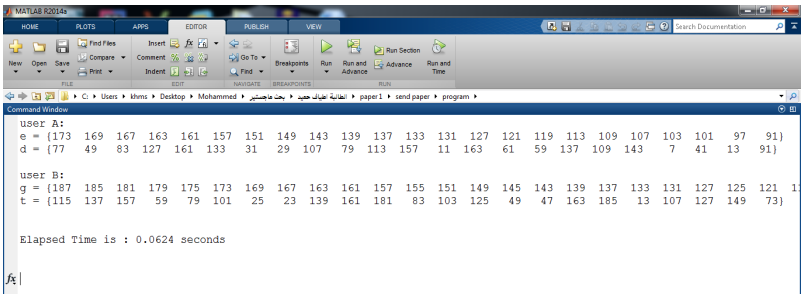

Figure 8: Multiple key generation for source and destination

4) The program implements the first scenario which user A consider as a source and user B consider as a destination.
5) The program selects the first session keys for user A and B as shown in Fig. 9 and then the program asks for entering the send message.

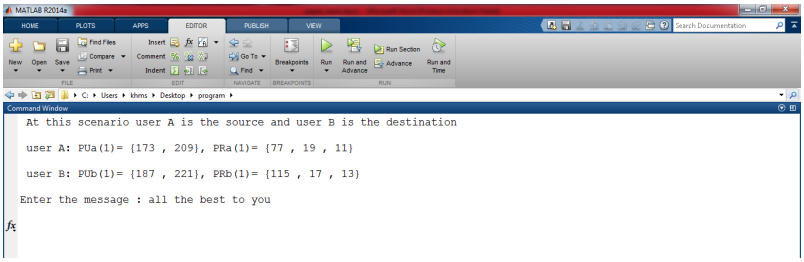

Figure 9: Selecting first session keys and entering the send message

6) At the source converting the message letters' to its corresponding ASCI code as shown in Fig. 10.
7) At the source user A the digital signature is computed by using PRa(1) according to equation 5.
8) After that at the source the output of step 7 is encrypted by using PUb(1) according to the equation 6.
9) At the destination (user B) the first process is decrypted the received message by using PRb(1) according to equation 7.
10) Finally, the receiver verify the digital signature of the sender by using PUa (1) according to equation 8 to achieve authentication and retrieve the original message, the elapsed time for the steps 6 - 10 is calculated as shown in Fig. 10.
11) Using the program RSA- MK.m for another scenario where user B now is the source and user A is the destination, the program choose the next set of session keys for the new scenario and at the source (user B) the message is entered as shown in Fig. 11.
12) Repeat the same steps 6 - 10 to verify the proposed MK- RSA algorithm with source user B and destination user A as shown in Fig. 12 From the results of the simulation can observe that the elapsed time for keys generation of SK- RSA algorithm is less than elapsed time for keys generation of Multi Keys RSA (M- RSA) as shown in Table I. While MK- RSA generates multiple keys in one implementation which are 23 keys pair (public and private) for each user, but SK- RSA one pair keys for each user in one implementation. Where MK- RSA algorithm deployed one session keys pair from the list of keys for each implementation to achieve the strategy of dynamic keys to ensure the security

of communication systems. In spite of the elapsed time of implementation of SK- RSA is less than MK- RSA as shown in Table II, but when compare the all times required for keys pair generation and implementation, MK- RSA is less than SK- RSA, where MK-RSA no need time for key generation in every communication session.

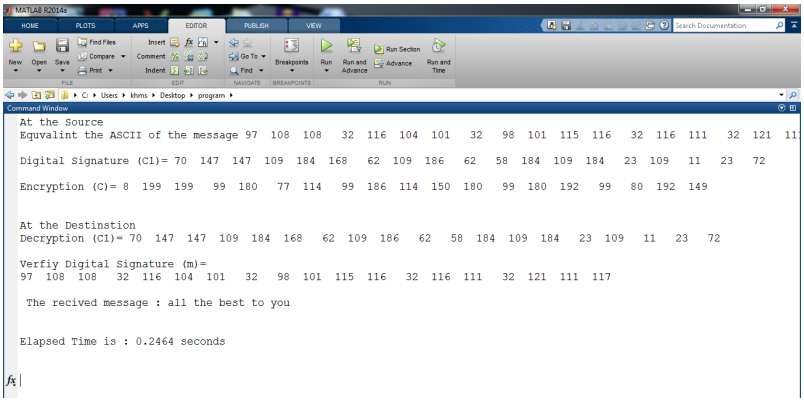

Figure 10: Computing digital signature and encryption at the source, and destination for MK- RSA

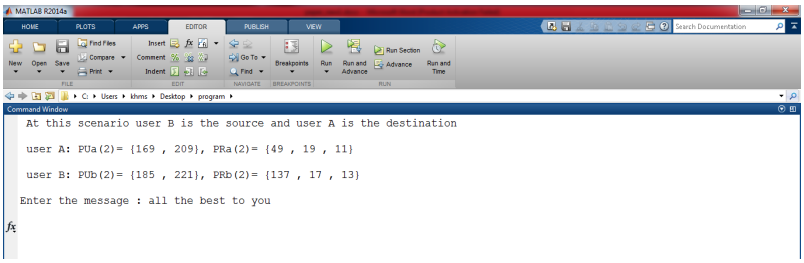

Figure 11: The program selecting the second session keys and entering the message

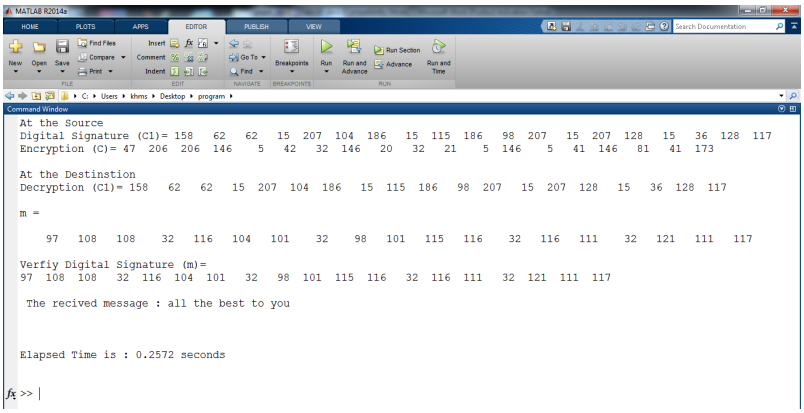

Figure 12: Computing digital signature and encryption at the source, and destination for MK- RSA

TABLE I
The Number of Generated Keys Pair and Their Elapsed Time

Algorithm name	Number of generated keys pair	The elapsed time of key generation (second)
SK- RSA	1	0.0468
MK-RSA	23	0.0624

TABLE II
The Elapsed Time of RSA Implementation of The Proposed Algorithm

Algorithm name	Scenario	The elapsed time of implementation (second)
SK- RSA	User A is the source and user B is destination	0.234
MK-RSA	User A is the source and user B is destination	0.2464
MK-RSA	User B is source and user A is destination	0.2572

V. CONCLUSIONS

This research proposed a new technique to enhance the security of RSA algorithm for providing message confidentiality and authentication. The results of the research show that the required computational time for key generation is increase for Multi Keys RSA (MK- RSA) when compared with the Single Key RSA (SK- RSA). But, SK- RSA in the next session, generates a new keys pair while the MK- RSA no need key generation, it uses the next keys pair in the list of the keys. Hence, all times required to execute the MK- RSA algorithm with multiple keys is less than the required time for SK- RSA. Also, the MK- RSA algorithm is more secure than SK- RSA algorithm with deploying the strategy of dynamic keys.

REFERENCES

[1] Islam, M. A, Islam, Md. A, Islam, N. and Shabnam, B. (2018) , " A Modified and Secured RSA Public Key Cryptosystem Based on " n " Prime Numbers" , Journal of Computer and Communications,Vol. 6, pp. 78- 90, https://doi.org/10.4236/jcc.2018.63006.
[2] S. A. Jaju, S. S. Chowhan, " A Modified RSA Algorithm to Enhance Security for Digital Signature" , International Conference and Workshop on Computing and Communication (IEMCON), Vancouver, 2015.
[3] Christof Paar, J. P, " Understanding Cryptography , Springer" , 2010.
[4] Dan Boneh, H. S, " Fast Variants Of RSA CryptoBytes" , 2002, pp. 1- 9.
[5] Rasha Samir Abdeldaym, H. M, (2019, march 6) , " Modified RSA Algorithm Using Two Public Key and Chinese Remainder Theorem" , I. J. of Electronics and Information Engineering , pp. 51- 64.
[6] Manoj Agrawal, B. L, (2015, December) , " Improvement Over Public Key Cryptosystem RSA by Implementing New Decryption Key Generation Algorithm" , International Journal of Engineering and Management Research , pp. 300- 304.
[7] R. D. Ardy et al , " Digital Image Signature using Triple Protection Cryptosystem (RSA, Vigenere, and MD5) " , International Conference on Smart Cities, Automation & Intelligent Computing Systems, Yogyakarta, Indonesia, 2017, pp. 08- 10.
[8] R. L. Rivest, A. S, (1978, february) , " A Method For Obtaining Digital Signatures Andpublic Key Cryptosystems" , Communicatiom of ACM , pp. 120- 126.
[9] I. Al-Barazanchi, et al , (2019, December) , " Modified RSA- Based Algorithm: A Double Secure Approach" , TELKOMNIKA [online] , Vol. 17, No. 6, pp. 2818- 2825.
[10] J. Quisquater, C. C, (1982, october 14), " Fast decipherment algorithm for RSA public- key cryptosystem" , ELECTRONICS LETTERS , pp. 905- 907.
[11] D. Mahto, et al, (June, 2019) , " Security Analysis of Elliptic Curve Cryptography and RSA" , Proceedings of the World Congress on Engineering [online], Vol. 1.
[12] D. Talukdar, L. P. Saikia , (2019, June) , " Simulation and Analysis of Modified RSA Cryptographic Algorithm using Five Prime Numbers" , International Journal on Recent and Innovation Trends in Computing and Communication [online] , Vol. 5, Issue. 6, pp. 224 - 228.
[13] A. K. Hussain, (2015, January) , " A Modified RSA Algorithm for Security Enhancement and Redundant Messages Elimination Using K- Nearest Neighbor Algorithm" , IJISET - International Journal of Innovative Science, Engineering & Technology [online] , Vol. 2, Issue. 1.
[14] Takagi, T, (1998, August) , " Fast RSA- type cryptosystem modulo p k q" , In Annual International Cryptology Conference (pp. 318- 326), Springer, Berlin, Heidelberg.
[15] Stallings, W. (2017) , " Cryptography And Network Security: Principles And Practice , Upper Saddle River: Pearson" .
[16] Hung Min Sun , Mu- En Wu , (2005) , " An Approach Towards Rebalanced RSA- CRT with Short Public Exponent" , IACR Cryptology ePrint Archive, https://eprint.iacr.org/2005/053.pdf.