RSA OVER-ENCRYPTION IMPLEMENTATION FOR NETWORKING: A PROOF OF CONCEPT USING MOBILE DEVICES

Walter Fuertes*, Fausto Meneses*, Luis Hidalgo*, Jenny Torres**
*Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador.
**Escuela Politécnica Nacional, Quito, Ecuador.

ABSTRACT

RSA is a cryptographic system which is widely used to protect the confidentiality of information transmitted over the Internet or other insecure networks. However, RSA is a probabilistic encryption algorithm, which has random components. This increases the probability that an attacker will find a technique for calculating the generated keys. To solve this problem, this study aims to increase the security of the RSA encrypted message by investing a minimal amount of extra time over-encryption and over-decryption. To achieve this, some variants have been introduced to the baseline method. Once the RSA encrypted message is obtained, a conversion of the decimal number system to a base b is performed. This results in a value of lesser length. The value n is generated randomly. To produce a mixture of the characters of the RSA message changed to base b, a matrix is used and the indexes of the rows of that matrix are randomly generated. In this case, each *cod* element is an ASCII character whose value is the position it occupies within its respective row. In practice the matrix has m rows per b columns where $m \in Z/1 < m < b!$. All rows represent strings randomly mixed. The results of the tests performed demonstrates an increase of 3.4 per thousand of time with respect to the decrypted-encryption RSA. Nevertheless, the algorithm greatly increases network security.

KEYWORDS: RSA, Security, Encryption, Over-encryption, Implementation.
MSC: 14G50, 68P25.

*Corresponding author: wmfuertes@espe.edu.ec

RESUMEN
RSA es un sistema criptográfico, que se usa ampliamente para proteger la confidencialidad de la información transmitida a través de Internet u otras redes inseguras. Sin embargo, RSA es un algoritmo de cifrado probabilístico, que tiene componentes aleatorios. Esto aumenta la probabilidad de que un atacante encuentre una técnica para calcular las claves generadas. Para resolver este problema, este estudio tiene como objetivo aumentar la seguridad del mensaje cifrado RSA invirtiendo una cantidad mínima de tiempo adicional en el cifrado y el descifrado. Para lograr esto, se han introducido algunas variantes en el algoritmo RSA clásico. Una vez que se obtiene el mensaje cifrado RSA, se realiza una conversión del sistema de números decimales a una base b. Esto da como resultado un valor de menor longitud. El valor de n se genera aleatoriamente. Para producir una mezcla de los caracteres del mensaje RSA cambiado a base b, se usa una matriz y los índices de las filas de esa matriz se generan aleatoriamente. En este caso, cada elemento cod es un carácter ASCII cuyo valor es la posición que ocupa dentro de su fila respectiva. En la práctica, la matriz tiene m filas por b columnas, donde m es un valor entero entre 1 y el $factorialde\ b$. Todas las filas representan cadenas mezcladas al azar. Los resultados de las pruebas realizadas demuestran un aumento de 3.4 por mil veces de tiempo con respecto al cifrado y descifrado RSA. No obstante, el algoritmo aumenta enormemente la seguridad de la red.

PALABRAS CLAVE: RSA, Seguridad, Encripción, Sobre-encripción, Implementación.

1. INTRODUCTION

The RSA encryption method was developed by its authors (Rivest, Shamir, Adleman) in 1977. RSA is an well-known algorithm and the most used worldwide. It provides security through the encryption of the information that surfs the Web, ensuring the confidentiality and authenticity of it [8]. This algorithm is very popular public key due to its simplicity in the calculation. However, the security of the RSA algorithm depends on the length of the prime numbers used for factoring. In addition, it is affected by the decomposition into prime factors, which for greater security requires greater length of the key, which implies an increase in computational cost [19] [20].
In this context, in order to increase the security level, the Multiple Encryption has been developed, which is the process of encrypting a message already encrypted one or more times, either using the same or another algorithm. There are many cryptographic schemes that combine redundant modules with the goal of achieving higher tolerance, increasing at the same time robustness and security [16]. The best known combiner is the cascade combiner (cascade or multiple encryption) applied to block ciphers and encryption schemes. Cascade encryption or multiple encryption, refers to systems that use several ciphers sequentially [16], providing an easy way of creating stronger symmetric cyphers encrypting a message multiple times by using different keys and/or different algorithms [4].
In this study, an RSA over-encryption algorithm is presented, which combines the modular and probabilistic calculation for encryption and decryption. To carry it out, the Unified Modeling Language (UML) and the Model-Driven Architecture (MDA) were applied. Next, we have analyzed the existing approaches to model encryption systems. With these results, we have designed a proposal for a model of encryption and decryption of information based on RSA. Finally, for its validation, a library has been implemented to encrypt the message in the client and send it together with the public keys

through the network; receive this data on the server and by accessing to the database, recover private keys, performing the decryption process to finally present the decrypted message to the recipient. To improve the safety of the model, the private keys are periodically updated through a mixing process. In addition, in order to measure the level of efficiency of the model, it has been compared with the traditional baseline algorithm, which works with the factorization for prime numbers of 300 digits also based on the RSA method [13].

Among the main contributions of this research we can mention: (1) the creation and application of a mathematical model that combines modular calculation with probabilistic calculation; (2) a matrix capable of generating messages about ciphers that could have information of equal value, but with different meaning; (3) a process of mixing and updating private keys; (4) message management through the use of mobile devices; (5) to convert a deterministic base project to a probabilistic project by generating random values; (6) To increase the security of the RSA encrypted message by investing a minimum amount of extra time in the encrypted and deciphered envelope (3.4 per thousand with respect to decrypted RSA encryption, according to the tests performed), which significantly increases the security of the network; (7) increases security by hiding private keys in executables and also in the encryption of RSA encryption.

The further part of this article has been organized as follows: Section 2 presents the related studies. In Section 3 the research design has been described as follows: a) First the approach of the problem is presented; b) Then we conduct a comparison between the model on over-encryption and the baseline; c) The design and implementation of the software used for this model is presented. Afterwards, in Section 4, we conducted a set of tests in order to demonstrate the results evaluation and discussion. Specifically, we performed a security test of the encryption algorithm to validate the security level. Finally, the conclusions and future studies have been presented in Section 5.

2. RELATED-WORK

Nowadays, this topic has received considerable attention from the standpoint of provable security. In the literature, different studies related with multiple encryption may be encountered [14]- [18]. For instance, Asmuth and Blakley [1] presented one of the first attempts with the creation of a cascade encryption applying a crypto-graphic algorithm. In the early 80s, it lacked of research of the involvement of the creation of stronger cryptosystems. Nevertheless, among the performed research at that time, Even and Goldreich [7] demonstrated that a cascade of block ciphers has been secure against message recovery attacks. Later on [5], Damgard and Knudsen proved that block ciphers cascade has been secure against chosen plain text key recovery attack. Finally in [12], Maurer and Massey documented that a cascade cipher is at least as strong as its first layer. They also revealed that a cascade may be weaker than the second layer of used encryption.

At present time, cascade encryption has been widely known and used. For instance, considering the different applications of multiple encryption, there has been an efficient multi-layer coding and encryption of MPEG video streams [17]. However, when dealing with multiple encryption, not only the double encryption (also called over-encryption), remain to be the only solution. The double encrypting provides a marginal increase in security, due to encountered in the middle attacks [6]. Bellare and

Rogaway [3] stated that a minimum of three iterations will be needed to provide a meaningful increase in security. Later on, it has been proved that longer cascades may also provide a meaningful increase in security, as long as the key is shorter than the plaintext and the number of iterations stay reasonable [9]. In practice, many cryptographic systems employ that approach. The best known solution has been the Triple DES [15], which uses cascade encryption applied to the DES block cipher. It applies the same cipher three times to increase key space and therefore security [15]. DES uses the encryption-decryption-encryption scheme (also known as EDE). Mennink and Preneel [14] presented a study about bridging the gaps of triple and quadruple encryption. In their study, triple encryption has been a cascade of three block cipher evaluations with independent keys, in order to enlarge its key size.
In a general form, in block cipher designs, we claim that each block cipher is a cascade cipher. All block ciphers have been composed of rounds of simpler ciphers, used with different round keys in order to compose a much stronger overall encryption [16]. In Feistel network ciphers and Lai-Massey ciphers [11][18] we observed that four rounds have been sufficient for the production of a strong random permutation. However, for the purposes of robustness, much larger numbers of rounds have been usually used. The throughput of such solutions is smaller, compared to the throughput provided by standard single encryption systems [16]. This allows them to be incompatible with mobile devices, due to the much higher computational and power costs.
When we compare the aforementioned studies with our current research, we realized instantly that most of them applied the DES and Triple DES algorithm. Nevertheless, using cascade encryption with RSA, we have achieved an optimized RSA which increases the security of the RSA, because the number of private keys has been able to be increased indefinitely through a matrix code creating much stronger symmetric key cryptographic algorithms.

3. RESEARCH-DESIGN

3.1. Problem Statement

Historically, in the application of RSA, the following issues have been identified: First, the mathematical solutions of RSA cryptographic algorithms, generate lengths of RSA encrypted messages to be quite large (in the order of 600 bytes), resulting in the consumption of greater amount of computational resources. Secondly, regardless of the length of the message, it has been required to store information encrypted in a database, which demands greater algorithmic and computational complexity. Thirdly, the increase of threats and vulnerabilities in computer networks, has motivated the obligation to desing an alternative, to improve the level of security of information, guaranteeing its availability, integrity and confidentiality. In order to solve these types of problems, our study aims to define a generic model that characterizes the optimization of the length of the encrypted message of the RSA method, with combined modular and probabilistic calculation with a set of private keys. Therefore, the problem to solve has been to design and build a generic model that meets all requirements and processes, specifically for RSA modeling.

3.2. Design and Implementation

Figure 1 illustrates the contextualization of this study. As it can be seen, a message is sent to the server from the transmitter client. In it, first, a simple encryption is conducted. On the server, the message is decrypted. This information is then sent to the receiver client, who through a local process, must over-decipher and present the original message to the user. Clients might be PCs, laptops, tablets, or smartphones.

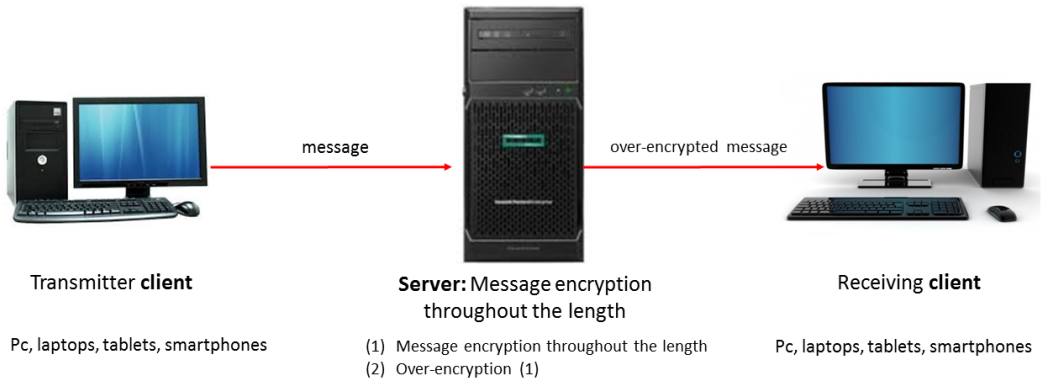


Figure 1: Over-Encryption Process.

As part of the application design, and according to Pressman Roger, a Class Diagram in the Design phase documents the specification for the software classes of an application. This includes classes, associations and attributes; Interfaces, with their operations and constants; Methods; Navigability; and Dependencies. In this study, the class diagram has been designed as illustrated in Fig. 2. In order to establish the interface between the JSF page and the server, the Message-Bean class has been designed, which is composed of the Message, MessageManager and MessageManagerLocal classes. In this case, the Message class encapsulates both the data and operations on the message, the over-encryption of the RSA, the decrypted message, the IP address of the remote client, the device from which the message is sent, among other parameters, while the MessageManager manages the information that has been sent and received from the client. MessageManagerLocal is an auxiliary class to send messages from the client. The Main class allows to control the process that has been performed on the server and has been composed of the classes Message, DataBase, BigInteger, Random1 and MeasureTime. In turn, the DataBase facilitates the storage and retrieval of information in the database. BigInteger contains the long integer value and its operations: encryption, over-deciphering, over-decryption and decryption of messages. Random1 is a class that generates random numbers that allow different encrypted messages to be produced in each execution of the application. Finally, MeasureTime performs the measurement of time in each stage of the encryption.

For the implementation of the software, a JSF project has been developed, capable of operating in both the desktop environment and the mobile device, based on the class diagram of Fig. 3. This figure indicates the GUI operating in a mobile environment. In the background, the server screen is visible and in the front of it a smartphone from which a message has been sent.

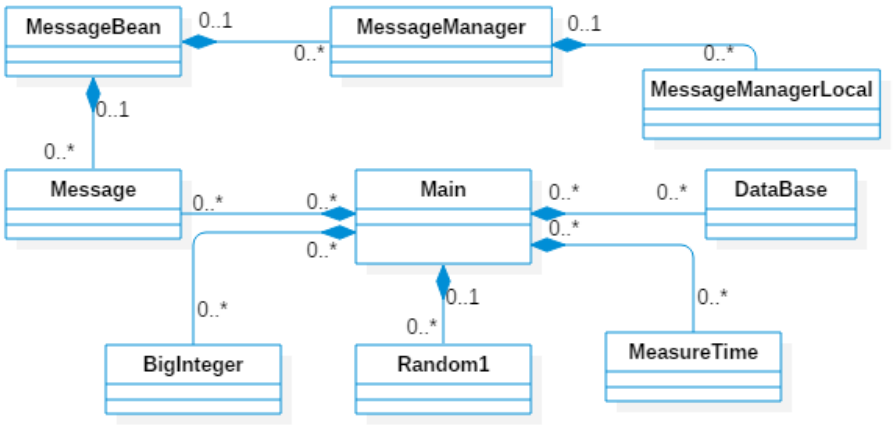


Figure 2: Class Diagram of the Application.

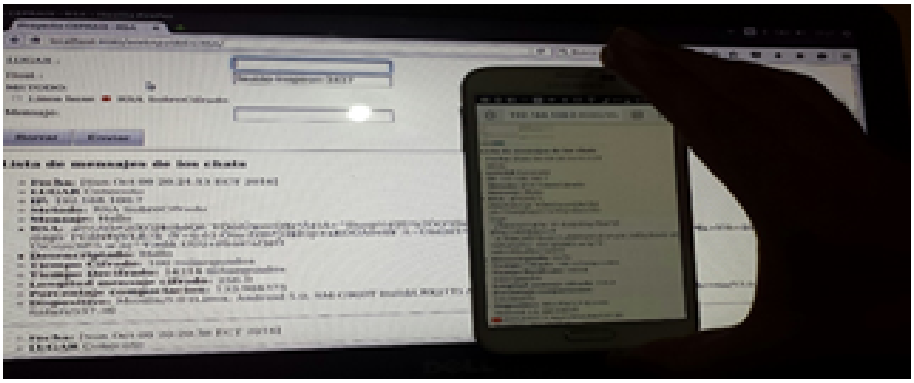


Figure 3: GUI of the RSA over-encryption model application

3.3. Analysis of security level

The main vulnerability presented by the baseline model has been that it allows an easy access to private keys (long prime primes) by editing the executable code (jar) of any implemented RSA program [13]. Another vulnerability has been due to the submission of the private keys to the sender, they have not been encrypted. In the RSA over-encrypted model, this situation does not occur, as all its private keys have been encrypted and decrypted during the creation of the encrypted message. Both fully-asymmetric encrypted messages have been generated in the base-line model and in the encrypted RSA. The proposed model increases significantly the security of the RSA, as the number of private keys have been able to be increased indefinitely through the matrix *cod*.

3.4. Comparative assesment

This subsection explains the algorithms of the baseline model that are taken as a reference to compare them with the RSA over-encrypted model, which are described below:

- RSA reference method
  The baseline RSA used in this work is based on the topic proposed by R. Johnsonbaugh [10], which established the procedure proposed by Meneses et al. [13].

- RSA encryption method
  It is based on introducing some variants to the previous method, these variants are:

  - Once the RSA encrypted message has been obtained, a conversion of the decimal numerical system to another of a base b is applied, for 200 <b <255. Thus, a lower length value is obtained.

  - The value n is generated randomly.

To produce a mixture of the characters of the RSA message changed to base b, a matrix (*cod*) (see Fig. 4) is used, and the indices of the rows of that matrix are randomly generated. In this case, each cod element is an ASCII character whose value is the position it occupies within its respective row. For example: assuming that the message changed to base b, produces the value I SEND YOU SIX HUNDRED DOLLARS; alf and *cod* are given by the attached table in Fig. 4. In addition, the indexes of the rows are generated in order 3, 5, 1, 4, and 2. Then the message about encryption is generated as RXAXENXRSNYNXYAD ALSOEAXHEIHY. The previous example is quite simple. In practice, the matrix has m rows per b columns where $m \in Z/1 < m < b!$. All rows represent randomly mixed chains.

3.5. Generation of public and private keys in the encryption

According to [13] and [2], encryption is the process of translating plain text data into something that appears to be random and meaningless (ciphertext). Within this study, these is the process: (1) Receive from the receiver the private keys (coded prime numbers of p and q and the matrix cod);(2) Decipher the prime numbers p and q; (3) Capture the message msj; (4) Generate a prime number n

Figure 4: GUI of the RSA over-encryption model application.

between 5 digits; (5) Randomly generate the ind indices of the rows of the matrix cod;(6) Calculate the ciphered message mjc from msj by applying the respective encryption process of the baseline method whose value is in decimal;(7) Calculate the overcrypted mjsc message by converting mjc to a base value b, mixing it using the matrix cod;(8) Send to receiver mjsc, n and ind;

3.6. Over-encryption process

When the message has been captured to be encrypted in the sender, the RSA encryption process generates a long base integer value 10. Next, the over-encryption process must be performed, which algorithm is detailed below. Then the encrypted message must be sent to the recipient, together with the public keys.

Algorithm 1 Over-encryption process pseudocode

Require: Inputs:

(1) cod is the code matrix for mixing;
(2) ind is the index vector of the rows of cod;
(3) mjc is the RSA encrypted message;
(4) nf is the number of rows of cod;
(5) b is the basis of the value on encryption;

Require: Process:

(1)Initialize message about imjsc encryption;
(2)Calculate i $<--$ 0;
(3) While mjc $>$ 0 do
(3.1) Calculate pos $<--$ Mod (mjc, b);
(3.2) Concatenate to imjsc the character of the pos position of cod in the row ind [Mod (i, nf)];
(3.3) Calculate mjc $<--$ mjc / b;
(3.3) Increase in one to i;
(4) End While;
(5) Return the inverted value of imjsc (that is, the overcrypted message mjsc);

The percentage of compaction can be calculated by applying the formula:
$c = (length(mjc) - length(mjsc)) * 100/length(mjsc)$.

3.7. Integral decryption process

Decryption is the process of converting ciphertext back to plaintext. This is the process: (1) Receive the public keys mjsc, n and ind from the sender; (2) Calculate the RSA mjc cipher base 10 message by converting base mjsc b, mixing it using the matrix cod; (3) Decipher the prime numbers p and q; (4) Calculate the message msj from mjc applying the respective decryption process of the baseline method; (5) Expand msj.

3.8. Over-decryption process

The process that describes the algorithm of previous section is conducted in the receiver. This consists of leaving the message received in a long integer value of base b. Finally, and through the algorithm process of section 3.8, it is converted to the sender's original message, ready to be accessed.

Algorithm 2 Over-decryption process pseudocode

Require: Inputs:

(1) cod is the code matrix for mixing;
(2) ind is the index vector of the rows of cod;
(3) mjsc is the message about RSA encryption;
(4) nf is the number of rows of cod;
(5) b is the basis of the mjsc;

Require: Process:

(1)Calculate the length lg of mjsc;;
(2) Calculate i < −− lg - 1;
(3) Calculate mjc < −− 0;
(4) Calculate power < −− 1;
(5) While i> = 0;
(5.1) Get the car character of the i position of mjsc;
(5.2) Calculate digit Position (cod [(ind [Mod (lg - i - 1, nf)], car);
(5.3)Increase the product between digits and power to mjc;
(5.4) Calculate power * b;
(5.5) Decrease in one to i;
(6) End While;
(7) Return mjc);

4. EVALUATION RESULTS

To validate our proposal in this research, we used the approximation method for numerical interpolation, where $k_1 = 0.041806020$; $k_2 = -0.0200291$ and $k_3 = -0.00687830180$. Polynomial interpolation,

approaching the third degree, results in the value of 597.98111292. In addition, the coefficients of the interpolation polynomial are obtained by means of a function written in Matlab. In the same way, we obtain the interpolation polynomial of: About Encryption Lenght Characters as a function of Length Msg Characters. For instance, Fig. 5 shows that the number of iterations in the deciphering process of the RSA baseline algorithm is 400 and 10000 times greater than the number of iterations that the decipherment of the over-encryption algorithm. This would means less use of computational resources.

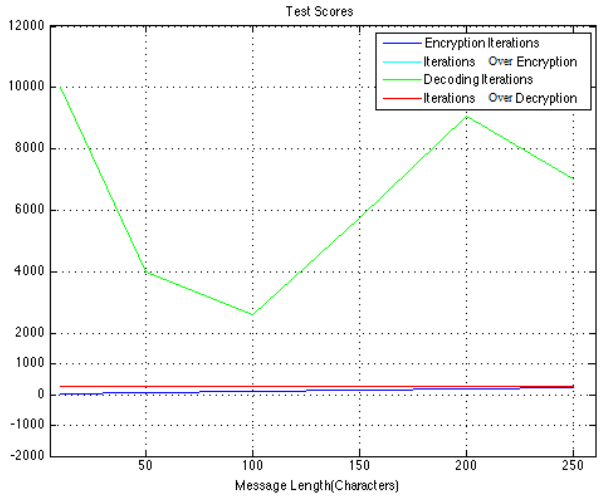


Figure 5: Number of iterations in the encryption and decryption process.

4.1. Security tests of the over-encryption algorithm

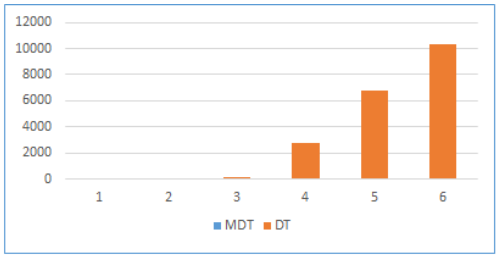
The traditional cipher formula is RSA $c = a.n \ mod \ f(p, q)$. Where c is the encrypted message, a is the message converted into numbers (i.e., coded), n is the exponent of the RSA encryption formula; c, a, p and q are long integer values. The security tests were conducted in order to establish the relationship between the probabilities of finding duplicates in the over-encrypted messages for the same message; where n is the exponent, *Sim* is the number of simulations and *rows* is the number of rows of the matrix. These three are independent variables, obtaining the following results:

Figures 6 and 7 represent the behaviour of the algorithm as a function of $Dig(n)$ (i.e., number of digits of n versus the following dependent variables: MDT = maximum dimension of the tuple; DT = number of duplicate tuples of the over-encrypted message; PC = percentage of compaction; OET = overclocking time in milliseconds; ODT = time of over-decryption in milliseconds.

As it can be seen, comparing between the 3-rows and the 4-row (Fig. 6 and 7, respectively) in Fig. 7, the values of the dependent variables (DT, PC, OET, ODT) decrease. As a result, the level of security of the algorithm increases, due to the reduction of the dimensions of the tuples and the quantity thereof.

Furthermore, from the results of the security tests, it can be deduced that as $Dig(n)$, Sim and

Security test: Dig(n)=6 and Rows=3			
Ind.	**Sim**	**MDT**	**DT**
1	1000	2	2
2	5000	2	29
3	10000	3	113
4	50000	2	2790
5	80000	3	6753
6	100000	2	10296



Security test: Dig(n)=6 and Rows=4			
Ind.	**Sim**	**MDT**	**DT**
1	1000	0	0
2	5000	2	9
3	10000	2	36
4	50000	2	751
5	80000	2	1864
6	100000	2	2984

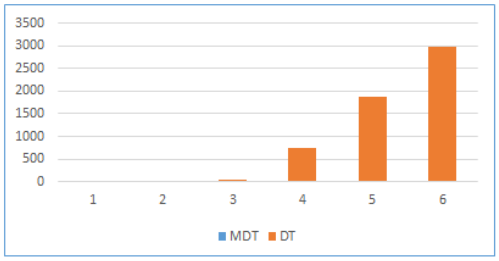
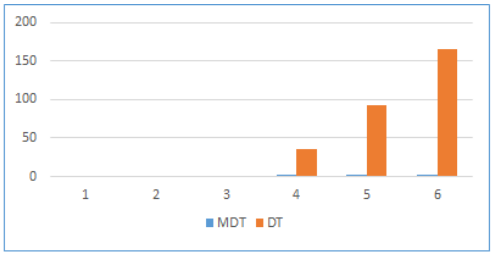


Figure 6: Initial experiment; number of simulations for $Dig(n) = 6$ and rows=3, 4.

rows increase, the number of duplicates of the over-encrypted message decreases. To be precise, the probability that duplicates of the over-encrypted messages will be found is directly proportional to Sim and inversely proportional to the function g $(Factorial(Dig(n)), Factorial(rows), C)$; where C is the quality of the random number generator. The rows of the matrix are private keys of the over-encryption process, which can increase indefinitely, constituting a strength for the security of the presented model.

Security test: Dig(n)=8 and Rows=3			
Ind.	**Sim**	**MDT**	**DT**
1	1000	0	0
2	5000	0	0
3	10000	0	0
4	50000	2	35
5	80000	2	93
6	100000	2	165



Security test: Dig(n)=8 and Rows=4			
Ind.	**Sim**	**MDT**	**DT**
1	1000	0	0
2	5000	0	0
3	10000	0	0
4	50000	2	11
5	80000	2	21
6	100000	2	36

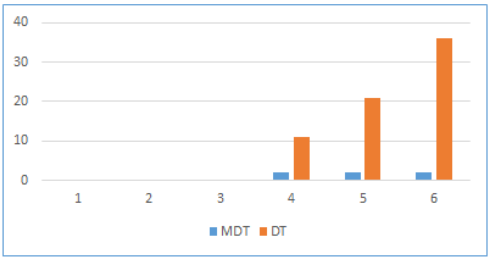


Figure 7: Final experiment; number of simulations for $Dig(n) = 8$ and rows=3, 4.

5. CONCLUSIONS

This research focused on optimizing the security of the RSA encrypted message using the technique called over-encryption and over-decryption. In order to achieve such goal, some variants have been introduced compared with previous models. Once the RSA encrypted message has been obtained, a conversion from the decimal number system to a base b is applied to $200<b<255$. This resulted in a value of shorter length. The value n has been generated randomly. We may point out, that for its implementation and validation algorithms have been modelled, which have been compared with the traditional baseline of the traditional RSA algorithm. The results of the performed tests evidence an increase in time, length, and iterations with respect to decrypted RSA. However, the algorithm greatly increases the security of the network. As future studies we have planned to extend this research in a DLL multiplatform.

RECEIVED: NOVEMBER, 2019.
REVISED: DECEMBER, 2019.

REFERENCES

[1] ASMUTH, C., AND BLAKLEY, G. (1981): An efficient algorithm for constructing a cryptosystem which is harder to break than two other cryptosystems. **Computers and Mathematics with Applications**, 7(6):447-450.

[2] AYALA, W., FUERTES, W., GALÁRRAGA, F., AULES H., AND TOULKERIDIS, T. (2017) "Software Application to Evaluate the Complexity Theory of the RSA and Elliptic Curves Asymmetric Algorithms," **2017 International Conference on Software Security and Assurance (ICSSA)**, Altoona, PA, 2017, pp. 87-93. DOI: 10.1109/ICSSA.2017.20.

[3] BELLARE, M., AND ROGAWAY, P. (2006). Code-Based Game-Playing Proofs and the Security of Triple Encryption, In: **Eurocrypt 2006. LNCS**, vol. 4004:409-426. Springer, Heidelberg.

[4] DAI, Y., LEE, J., MENNINK, B., AND STEINBERGER, J. (2014): The security of multiple encryption in the ideal cipher model. **In International Cryptology Conference**. Springer, Berlin.

[5] DAMGARD, I., AND KNUDSEN, L. (1994, November): Enhancing the strength of conventional cryptosystems.

[6] DIFFIE, W., AND HELLMAN, M. E. (1977): Exhaustive Cryptanalysis of the Data Encryption Standard. **Computer** 10:74-84.

[7] EVEN, S., AND GOLDREICH, O. (1985): On the power of cascade ciphers. **ACM Transactions on Computer Systems (TOCS)**, 3(2):108-116.

[8] FARAH, S., JAVED, Y., SHAMIM, A., AND NAWAZ, T. (2012): An experimental study on performance evaluation of asymmetric encryption algorithms. **In Recent Advances in Information Science, Proceeding of the 3rd European Conf. of Computer Science,(EECS-12)**. 2012.

[9] GAZI, P., AND MAURER, U. (2009): Cascade Encryption Revisited ETH Zurich, Switzerland Department of Computer Science, Comenius University, Bratislava, Slovakia Department of Computer Science.

[10] JOHNSONBAUGH, R. (2005): Matemáticas discretas. Pearson Educación, México, ISBN: 970-26-0637-3.

[11] LUBY, M., AND RACKOFF, C. (1988): How to construct Pseudorandom Permutations from pseudorandom functions, **SIAM Journal on Computing**, 17(2):373-386.

[12] MAURER, U., AND MASSEY, J. (1993): Cascade ciphers: The importance of being first. **Journal of Cryptology**, 6(1):55-61.

[13] MENESES, F., FUERTES, W., SANCHO, J., SALVADOR, S., FLORES, D., AULES, H., AND NUELA, D. (2016): RSA Encryption Algorithm Optimization to Improve Performance and Security Level of Network Messages. **International Journal of Computer Science and Network Security (IJCSNS)**, 16.8(2016):55.

[14] MENNINK, B., AND PRENEEL, B. (2014): Triple and Quadruple Encryption: Bridging the Gaps, Dept. Electrical Engineering, ESAT/COSIC, KU Leuven, and iMinds, Belgium

[15] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (2004): Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST Special Publication 800-67.

[16] PAMUKOV, M. E., POULKOV, V., MIHOVSKA, A., PRASAD, N. R., AND PRASAD, R. (2014): Lightweight robust cryptographic combiner for mobile devices: Crypto roulette. **2014 IEEE 19th CAMAD**, Athens, 2014:188-192.

[17] TOSUN, A. S., AND FENG, W. C. (2000): Efficient multi-layer coding and encryption of MPEG video streams. **In Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on**, New York, NY, 2000, pp. 119-122 vol.1.

[18] VAUDENAY, S. (1999): On the Lai-Massey Scheme, **Lecture Notes in Computer Science** Volume 1716:8-19.

[19] VERMA, S., AND GARG, D. (2014): An Improved RSA Variant. **International Journal of Advancements in Technology**, 5(2):161-169.

[20] YADAV, P. S., SHARMA, P., AND YADAV, K. P. (2012). Implementation of RSA algorithm using Elliptic curve algorithm for security and performance enhancement. **International Journal of Scientific and Technology Research**, 1.4(2012):102-105.