Enhanced, Modified and Secured RSA Cryptosystem based on *n* Prime Numbers and Offline Storage for Medical Data Transmission via Mobile Phone

Achi Harrisson Thiziers[1*], Haba Cisse Théodore[2], Jérémie T. Zoueu[3], Babri Michel[4]
Instrumentation, Imaging and Spectroscopy Laboratory (L2IS)
Institut National Polytechnique-Houphouët Boigny (INP-HB)
DFR-GEE, Yamoussoukro, Côte d'Ivoire[1,2,3]
Computer Science and Telecoms Networks Laboratory (LARIT)
INP-HB Abidjan, Côte d'Ivoire[4]

***Abstract*—The transmission of medical data by mobile telephony is an innovation that constitutes the m-health or more generally e-health. This telemedicine handles personal data of patients who deserve to be protected when they are transmitted via the operator or private network, so that malicious people do not have access to them. This is where cryptography comes in to secure the medical data transmitted, while preserving their confidentiality, integrity and authenticity. In this field of personal data security, public key cryptography or asymmetric cryptography is becoming increasingly prevalent, as it provides a public key to encrypt the transmitted message and a second private key, linked to the first by formal mathematics, that only the final recipient has to decrypt the message. The RSA algorithm of River and Shamir provides this asymmetric cryptography based on a public key and a private key, on two prime numbers. However, the factorization of these two prime numbers to give the variable *N* of RSA can be discovered by a hacker and thus make the security of medical data vulnerable. In this article, we propose a more secured RSA algorithm with *n* primes and offline storage of the essential parameters of the RSA algorithm. We performed a triple encryption-decryption with these *n* prime numbers, which made it more difficult to break the factorization of the variable *N*. Thus, the key generation time is longer than that of traditional RSA.**

Keywords—e-Health; medical data transmission; asymmetric cryptography; RSA algorithm; first numbers

I. INTRODUCTION

Transmitting medical data via interconnection technologies such as mobile telephony is an operation that requires the highest level of security, in order to preserve their private and personal nature. This subject, as well as the algorithm of River and Shamir [1], have been the topic of several studies in the literature and continue to fascinate many researchers. D. Sathya and al. [2] worked on a secure remote monitoring system, combining a symmetric algorithm and attribute-based encryption, to secure data transmission and the medical sensor network access control system. J. Heurix and al. [3] have worked on storage that preserves privacy and access to medical data through pseudonymization and encryption. Mohammed L. and al. [4] worked on remote supervision of e-health that preserves privacy, through a process of prior patient approval, before any transmission to the Health Centre. M. Milutinovic and al. [5] spoke about the management of privacy-preserving data in an e-health system, developing a protocol based on new e-health architecture.

We note from these works that data encryption aims to make medical data inaccessible to unauthorized persons. Thus, the confidentiality, integrity and availability of this data are preserved [6]. There are two main types of cryptography. Symmetric key cryptography, with a unique public key that is shared between the sender who sends the encrypted message and the receiver who receives it and decrypts the full text. Among the symmetric algorithms are DES, 3DES, AES, IDEA, and BLOWFISH [7]. We also have asymmetric cryptography with two distinct keys: a public key that the sender uses to encrypt his message and another private key mathematically linked to the first that is used to decrypt the original message. We can mention here the RSA algorithm which factors two prime numbers to give a large integer number 'N' [8]. The simple principle that drives RSA is to be able to perform easy mathematical calculations, but whose reverse operation is difficult, in the absence of additional information, according to M. A. Islam and al. [9].

In general, RSA uses two primes "p" and "q" to obtain the factorization of the large integer "N". The attack on RSA can occur at this level, when the hacker succeeds in discovering the factorization of the large number "N", thus preventing the generation of the private key from the public key.

Our contribution, in this article, is an amelioration of the security of RSA, by accentuating key generation time, during the factorization of the large number *N*. We used, like M. A. Islam, four prime numbers, instead of two, in the original RSA model; this makes the factorization more robust with a large number of the exponent used for encryption. Instead of the double encryption-decryption he performed, we made a triple encryption-decryption to make RSA even stronger, therefore more secured than the original RSA of Shamir and MSRSA from Muhammad. To speed up encryption-decryption, we stored offline the essential key generation and factorization parameters.

The first part of this article, constituted by the introduction, is followed by the second part which relates the state of the art,

*Corresponding Author

in terms of cryptosystems based on the modification of the RSA Algorithm. In the third part, we present the method and material of our research. The original RSA algorithm, Muhammad Ariful's Modified and Secured RSA algorithm and our reinforced, modified and secured RSA Algorithm are presented in Parts Four and Five respectively. Part Six presents the implementation and results. As for the fifth part, it gives a conclusion and suggests avenues for our future works.

II. Related Works

Several research studies in doctoral theses and scientific journals [32-36], have shown that it is possible to improve the security of encryption and decryption of personal and private data. Concerning the RSA algorithm, this work is abundant, and in this literature review, we present a non-exhaustive table of this work. H. Ali and al. [10] proposed an amendment to RSA called timing attack prospect for RSA cryptanalysts using genetic algorithm technique. This article proposes the use of a genetic algorithm to measure the time required to attack the crypto RSA system. B. Kumar and al. [11] proposed a hybridization of the AES and RSA algorithm for clouds. A. Bhardwaja and al. [12] presented security algorithms for cloud computing. B. Swamia and al. [13] proposed an algorithm based on a double modulo at the RSA algorithm using the Jordan-Totient function. Dr. P. Mahajan and al. [14] reviewed the literature on encryption based on AES, DES and RSA algorithms for data security. D. Preuveneersa and al. [15] wrote an article on the future of mobile telephone-based e-health application development, examining HTML5 for diabetes management in an intelligent environment. M. Kethari and al. [16] produced a literature review on the transmission of medical data for e-health in terms of web platform security. V. Kapoor and al. [17] have produced a new hybrid cryptography technique to consolidate network security. K. G. Kadam and al. [18] also produced a hybrid algorithm using both RSA-AES encryption for web services. K. Rege and al. [19] also used the hybridization of AES and RSA algorithms to secure Bluetooth communication. R. Raj and al. [20] worked on the modification of the crypto RSA system. S. Patel and al. [21] have implemented a new encryption method using a modification of the RSA algorithm and the Chinese recall theorem. A. Gupta and al. [22] examined a double modification of the modulo of the RSA algorithm and tested it with a brute force attack. B. Yüksel and al. [23] have produced research on privacy and security of electronic services. S. Bhuyan and al. [24] wrote an article on privacy and security issues in mobile health: Current research and future directions. H. S. G. Pussewalage and al. [25] published on privacy mechanisms for enforcing security and privacy requirements in e-health solutions. Y. Li and al. [26] worked on the design and implementation of an improved RSA algorithm. S. Sharma and al. [27] produced a new variant of the RSA subset-sum cryptosystem. Amare Anagaw Ayele and al. [28] have implemented a modified RSA encryption technique based on multiple public keys. H. Huang and al. [29] wrote an article on the transmission and analysis of private and secure medical data for a wireless health care system with detection. B. P. U. Ivy and al. [30] published an article on a modified RSA encryption system based on 'n' prime numbers.

All these works have shown that it is possible to improve the safety of the RSA Algorithm by modifying it and enhancing its safety, and even more so by making it faster. Indeed, one of the weaknesses of RSA is the relatively long time for the execution of the algorithm. Our research consists in strengthening the generation of the private key, by using several prime numbers, and by proceeding with triple encryption-decryption. This increases security, as the generation of the private key takes more time and makes it more difficult to factorize the large number of 'N', which the Hacker will have difficulty breaking easily. To make the algorithm run faster, in addition to the triple encryption-decryption, we stored in a database the essential parameters for generating the private key.

III. Materials and Method

The data to be secured are the ones transmitted by mobile phone to a cloud, to enable e-health, protecting patients' private data, as shown in Fig. 1. Our device is a multi-sensor called "6 in 1 Health monitor" that allowed us to acquire directly on the patients, 6 health constants: Infrared temperature, blood pressure, blood sugar, blood oxygen saturation level SpO2, ECG electrocardiogram, and heart rate. These constants are then transmitted to the mobile application by Bluetooth, and then, the tablet transmits them via the mobile application we have developed, on the telephone network, to a special doctor who can interpret them and give the necessary recommendations.

To achieve this objective, we propose, as in the work of Muhammad and al. a modification of RSA. However, unlike the pair of random numbers he used, we use a four random numbers, with their inverse multiplication module, to further increase the security of RSA. We have obtained a greater generation time of the private key for MRSA of Muhammad and therefore also for the original RSA. The encryption and decryption time is much longer than that of the original RSA, but we were able to obtain that, despite the triple encryption and decryption, the encryption and decryption time of our EMSRSA is almost equal to that of MSRA, by storing offline the key parameters for generating the private key in a database. This results in faster execution of encryption and decryption. Better than Muhammad and al., we implemented a real simulation interface, which we called 'RSA GENERATOR', to achieve performance tests. This interface has been implemented in a JAVA environment; we have used, like Muhammad and al., the functions of the same large integer library. As Muhammad and al. said, this library offers several functions such as modular arithmetic, calculation of the highest common denominator, primacy test, prime number generation, bit manipulation and many other operations. For EMSRSA, Via 'RSA GENERATOR', based on our reference article, the user has the possibility to choose prime numbers or to choose the size of the bits in order to automatically generate the prime numbers. Subsequently, a comparison of the execution times of the private key generation time, the encryption time and the decryption time is made between the original RSA, MSRA and EMSRSA.

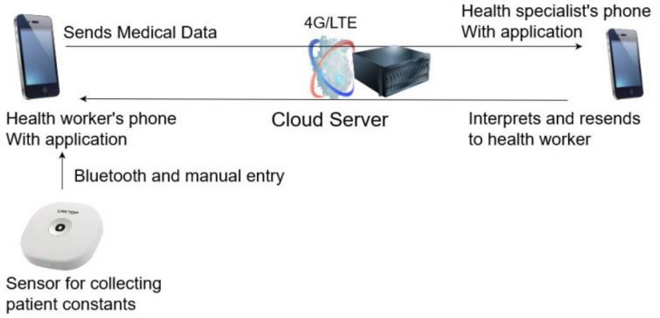


Fig. 1. Overview of our Mobile Cloud Computing for Secure Transmission of Medical Data.

A. Functionning of the original RSA Algorithm of Shamir and al.

Private Key generation

a) Select p and q, both prime numbers, p being different from q.

b) Calculate

$$n = p * q \tag{1}$$

c) Calculate the Euler phi value of n

$$Ø(n) = (p -1) * (q-1) \tag{2}$$

d) Randomly select an integer *e* that satisfies the following conditions

$$\text{PGCD } (Ø\ (n), e) = 1 \tag{3}$$

Where

$$1 < e < Ø(n) \tag{4}$$

e) Calculate a random number d such that:

$$d = e\text{-}1 \text{ (mod } Ø\ (n)) \tag{5}$$

Or

$$d * e \equiv 1 \text{ (mod } Ø\ (n)) \tag{6}$$

f) Public Key

$$PUK = \{e, n\} \tag{7}$$

g) Private Key

$$PRK = \{d, n\} \tag{8}$$

Encryption

ncryption is done using the public key: {*e*, *n*}

a) The full text - Message (M)

b) The encrypted text:

$$C = \text{M}^e \text{ mod } n \tag{9}$$

Decryption

The decryption is done by the private key: {*d*, *n*}

a) Encrypted text - C

b) Full text

$$M = \text{C}^d \text{ mod } n \tag{10}$$

Where *M* is the original message, *p* and *q* being prime numbers; *n* is their common modulo, *e* and *d* being respectively the public and private key.

B. Functioning of the MSRSA Algorithm of Muhammad Islam and al.

Private Key generation

a) Select *p* and *q*, *r*, *s*, all four of the prime numbers, *p*, *q*, *r*, *s* being different.

b) Calculate:

$$n = p * q * r * s \tag{11}$$

c) Calculate the Euler phi value of n:

$$Ø(n) = (p -1) * (q\text{-}1) * (r\text{-}1) * (s\text{-}1) \tag{12}$$

d) Randomly select two integers *e* and *f* that meet the following conditions.

$$\text{PGCD } (Ø\ (n), e) = 1 \tag{13}$$

Where

$$1 < e < Ø(n) \tag{14}$$

$$\text{PGCD } (Ø\ (n), f) = 1 \tag{15}$$

Where

$$1 < f < Ø(n) \tag{16}$$

e) Calculate a random number *d* such that:

$$d = e \text{ -1 (mod } Ø\ (n)) \tag{17}$$

Or

$$d*e\equiv 1 \text{ (mod } Ø\ (n)) \tag{18}$$

Calculate another random number *g* so that:

$$f * g \equiv 1 \text{ (mod } Ø\ (n)) \tag{19}$$

f) Public Key

$$PUK = \{e, f, n\} \tag{20}$$

g) Private Key

$$\text{PRK} = \{d, g, n\} \tag{21}$$

Encryption

Encryption is done by public key: {*e*, *f*, *n*}

a) Encrypted text - C

b) Full text:

$$C = ((\text{M}^e \text{ mod } n)^f) \text{ mod } n) \tag{22}$$

Where M is the original message, *p* and *q*, *r*, *s* being prime numbers. ‘n' is their common modulo, (*e*, *f*) and (*d*, *g*) being respectively the public and private key.

Decryption

The decryption is done by the private key: {*d*, *g*, *n*}

a) Encrypted text - C

b) Full text:

$$M = ((\text{C}^g \text{ mod } n)^d) \text{ mod } n) \tag{23}$$

Where M is the original message, p and q, r, s being prime numbers. ‘n’ is their common modulo.

C. Functioning of our Proposed EMSRSA Algorithm

Private Key generation

a) Select p and q, r, s, all four of the prime numbers, p, q, r, s being different.

b) Calculate:

$$n = p * q * r * s \tag{24}$$

c) Calculate the Euler phi value of n:

$$Ø\,(n) = (p\text{ -}1) * (q\text{-}1) * (r\text{-}1) * (s\text{-}1) \tag{25}$$

d) Randomly select integers e, f, h, i that meet the following conditions:

$$\text{PGCD}\,(Ø\,(n), e) = 1 \tag{26}$$

Where

$$1 < e < Ø(n) \tag{27}$$

$$\text{PGCD}\,(Ø\,(n), f) = 1 \tag{28}$$

Where

$$1 < f < Ø(n) \tag{29}$$

$$\text{PGCD}\,(Ø\,(\text{n}), h) = 1 \tag{30}$$

Where

$$1 < h < Ø(n) \tag{31}$$

e) Calculate a random number d such that:

$$d = e\text{ -}1\ (\text{mod } Ø\,(n)) \tag{32}$$

Or

$$d * e \equiv 1\ (\text{mod } Ø\,(n)) \tag{33}$$

Calculate two other random numbers g and i, so that:

$$(f * g) * (h * i) \equiv 1\ (\text{mod } Ø\,(n)) \tag{34}$$

f) Public Key:

$$PUK = \{e, f, h, n\} \tag{35}$$

g) Private Key

$$PRK = \{d, g, i, n\} \tag{36}$$

Encryption.

Encryption is done by public key: $\{e, f, h, n\}$

a) Encrypted text - C

b) Full text

$$M = ((((\text{M}^{e} \text{ mod } n)^{f}) \text{ mod } n)^{h}) \text{ mod } n) \tag{37}$$

Where M is the original message, p and q, r, s being prime numbers. ‘n’ is their common modulo, (e, f, h) and (d, g, i) being respectively the public and private key.

Decryption.

The decryption is done by the private key: $\{d, g, i, n\}$

a) Encrypted text - C

b) Full text

$$M = ((((\text{C}^{g} \text{ mod } n)^{d}) \text{ mod } n)^{i}) \text{ mod n}) \tag{38}$$

Where M is the original message, p and q, r, s being prime numbers; ‘n’ is their common modulo.

Fig. 2 shows a flow diagram of our Enhanced Modified and Secured RSA (EMSRSA) algorithm. Like M. Islam and al., we used four prime numbers to calculate n and Ø(n). This time, a triplet of numbers (e, f and h) obtained at random is used in the range $1< e < Ø(n)$, $1< f < Ø(n)$ and $1< h < Ø(n)$ as exponent of the public key. Subsequently, the inverse modulo multiplication of these random numbers (d, g and i) is calculated to serve as a private key exponent. Encryption and decryption are carried out by the exponents of this public and private key.

(MRSA) of M. A. Islam and al.

As with Muhammad's article, let's look at a concrete example using, this time, our EMSRSA.

Let’s take four prime numbers: p = 53, q = 41, r = 43, s = 47.

We Calculate:

$$n = p*q*r*s$$

$$n = 53*41*43*47=4391633$$

We compute Euler phi value of n:

$$Ø(n) = (p\text{-}1) * (q\text{-}1) * (r\text{-}1) * (s\text{-}1)$$

$$Ø(n) = (53\text{-}1) * (41\text{-}1) * (43\text{-}1) * (47\text{-}1) = 4018560$$

We randomly select two integers e and h that meet the following conditions.

PGCD $(Ø\,(n), e) = 1$ and $1< e < Ø(n)$ give $e = 41$

PGCD $(Ø\,(n), h) = 1$ and $1 < h < Ø(n)$ give $h = 53$

We randomly select an integer d that meets the following conditions.

$$d * e \equiv 1 \text{ mod } Ø(n)$$

We deduce

d = 294041.

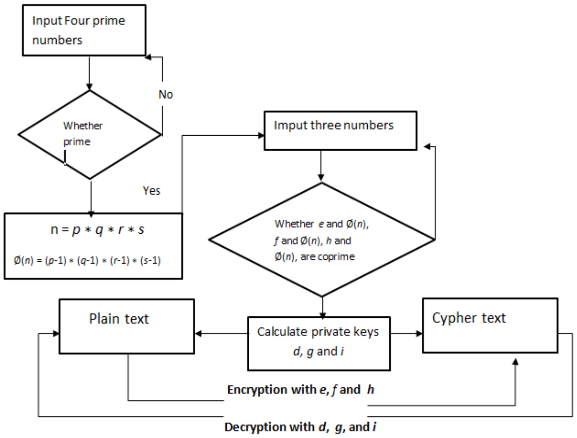

Input Four prime numbers
No
Whether prime
Yes
n = p ∗ q ∗ r ∗ s
Ø(n) = (p-1) ∗ (q-1) ∗ (r-1) ∗ (s-1)
Imput three numbers
Whether e and Ø(n), f and Ø(n), h and Ø(n), are coprime
Plain text
Calculate private keys d, g and i
Cypher text
Encryption with e, f and h
Decryption with d, g, and i

Fig. 2. Flow Diagram of our Algorithm (EMSRSA), Inspired by.

We randomly select another integer f that meets the following conditions

PGCD $(\text{Ø}(n), f) = 1$ and $1 < f < \text{Ø}(n)$ give $f = 97$.

We randomly compute an integer g that meets the following conditions

$g * f \equiv 1 \bmod \text{Ø}(n)$

We deduce

$g = 455713$

We randomly select an integer i that meets the following conditions

$(i * h) \equiv 1 \bmod \text{Ø}(n)$

And

$(g * f) * (i * h) \equiv 1 \bmod \text{Ø}(n)$

Input Message is M=12321

Let's Encrypt the Message

$C = ((((M^{e} \bmod n)^{f}) \bmod n)^{h}) \bmod n) = 3475386$

We then decrypt the crypted Message C

$M = ((((C^{g} \bmod n)^{d}) \bmod n)^{i}) \bmod \text{n}) = 12321$

D. Implementation

Our algorithm implementation was done in JAVA version OXYGEN environment on a Lenovo computer, Intel® Core TM i5 CPU M520 (2.40 GHz)*(2.40 GHz), with 8 GB of RAM. The security challenge of the RSA Algorithm and other modified algorithms such as Muhammad's is to consolidate the factorization of the 'n' number in order to increase the difficulty of detecting the key by a hacker.

IV. Results and Discussion

A. Performance Analysis

To compare the primary key generation times, encryption and decryption times of the three algorithms, we used the same bit sizes in our RSA Generator interface. Table I, Table II and Table III give performance of the RSA, MRSA and our EMSRSA Algorithms.

Fig. 3 shows our developed interface called 'RSA GENERATOR'. We used it to test both, RSA, MRSA and our EMSRSA performance.

Fig. 4 is the graph of Key generation time comparison for the three algorithms.

Fig. 5 and Fig. 6 following are respectively the graph of encryption time comparison and the graph of decryption time comparison for the three algorithms.

For prime number entries for the 2048 size bit, our key generation time is 21.220416 ms compared to 11.668641 ms for Muhammad's MRSA. This proves that our enhanced EMSRA is stronger than MRSA and RSA. Our encryption time is longer with our method (1285.051478 ms) compared to 16.20906 ms for MRSA, but our decryption time is shorter (2699.154794 ms) than (2809.571451 ms) for MRSA.

B. Complexity Analysis

Complexity of the RSA algorithm

As Muhammad showed for RSA, the complexity of the two randomly selected numbers is.
$O\ (s* (\log2p)^{2} *\ln p)$ and $O\ (s* (\log2q)^{2}*\ln q)$, due to MILLER RABIN complexity.
The complexity of randomly finding the variable 'e' being
$O\ ((\log2(\log2p-1) * (\log2q-1))^{2}+1)$.
Similarly for our EMSRSA, as for the MRSA of Muhammad, the complexity to find the prime numbers p, q, r, s that we will note for the occasion w, x, y, z is respectively:
$O\ (s* (\log2w)^{4} *\ln w)$, $O\ (s* (\log2x)^{4} *\ln x)$,
$O\ (s* (\log2y)^{4} *\ln y)$ et $O\ (s* (\log2z)^{4} *\ln z)$.
The complexity for the hacker to find the numbers at random e, f, and h being the same:
$O\ ((\log2(\log2w-1)*(\log2x-1)*(\log2y-1)*(\log2z-1))^{4}+1)$.

We can see that the EMSRSA Algorithm is more complex than the RSA algorithm.

TABLE. I. ORIGINAL RSA PERFORMANCE

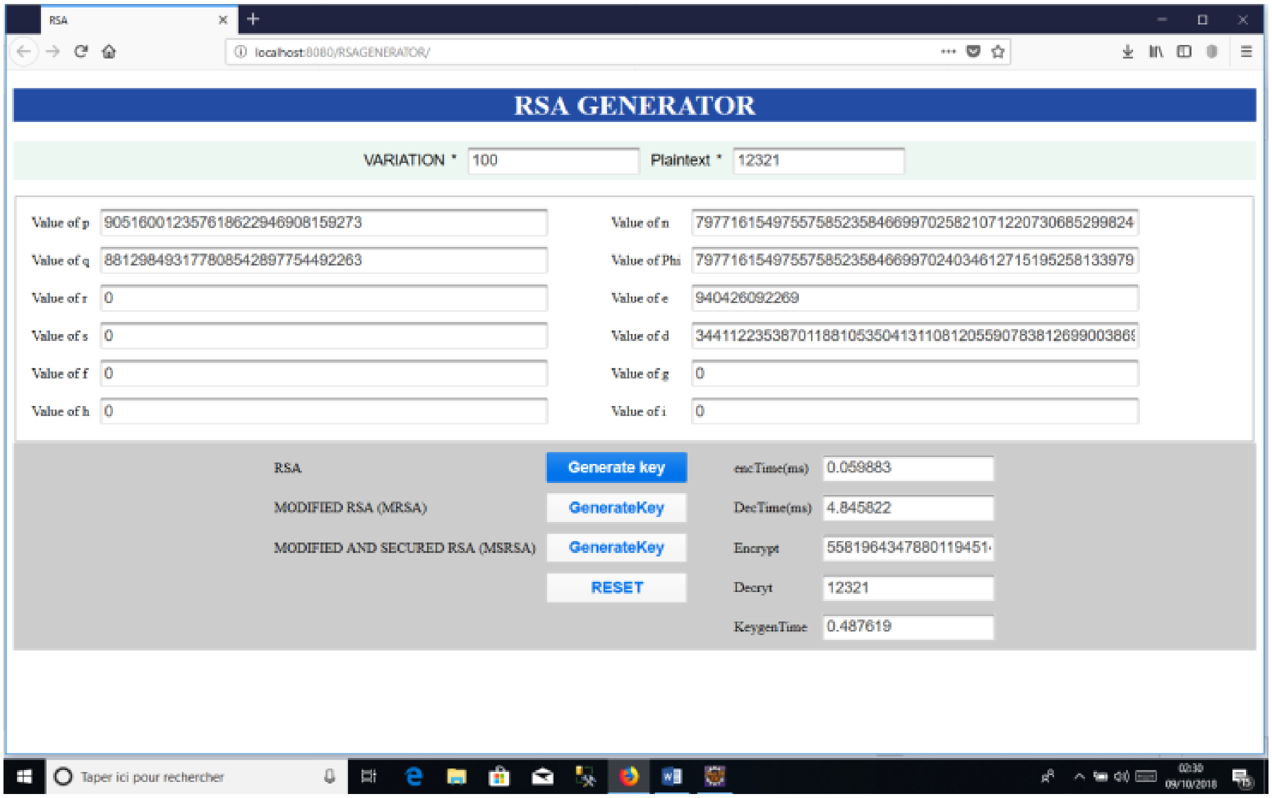
Length of p, q (byte)	Key Generation time (in ms)	Encryption time (in ms)	Decryption time (in ms
100	0,487619	0,059883	4,845822
128	0,397367	0,0586	0,257925
256	0,536381	0,102229	0,873438
512	0,583859	0,151419	3,477494
1024	1,022289	0,962406	24,465223
2048	1,878189	2,863266	167,77436
4096	6,547785	10,178836	1464,092836

TABLE. II. PERFORMANCE OF MRSA

Length of p, q,r,s (byte)	Key Generation time (in ms)	Encryption time (in ms)	Decryption time (in ms
100	1,150182	0,155696	1,250701
128	0,948718	0,169812	1,941494
256	2,089063	0,506868	15,558901
512	3,352168	1,804191	53,716385
1024	3,724298	4,027991	337,387986
2048	11,668641	16,20906	2809,571451
4096	43,781358	87,732529	20253,26726

TABLE. III. PERFORMANCE OF OUR PROPOSED EMSRSA

Length of p, q,r,s (byte)	Key Generation time (in ms)	Encryption time (in ms)	Decryption time (in ms
100	2,262297	0,71603	1,193811
128	1,846109	0,747683	1,176702
256	3,756807	10,668167	14,275265
512	7,11368	33,898088	51,301388
1024	12,052748	199,573975	377,112268
2048	21,220416	1285,051478	2699,154794
4096	65,98343	10237,78236	20158,48051



RSA GENERATOR

VARIATION * 100 Plaintext * 12321

Value of p	905160012357618622946908159273	Value of n	797716154975575852358466997025821071220730685299824
Value of q	881298493177808542897754492263	Value of Phi	797716154975575852358466997024034612715195258133979
Value of r	0	Value of e	940426092269
Value of s	0	Value of d	3441122353870118810535041311081205590783812699003869
Value of f	0	Value of g	0
Value of h	0	Value of i	0

RSA

MODIFIED RSA (MRSA)

MODIFIED AND SECURED RSA (MSRSA)

encTime(ms) 0.059883

DecTime(ms) 4.845822

Encrypt 5581964347880119451

Decryt 12321

KeygenTime 0.487619

Fig. 3. JAVA Interface 'RSA GENERATOR' for a Size bit Set to 100.

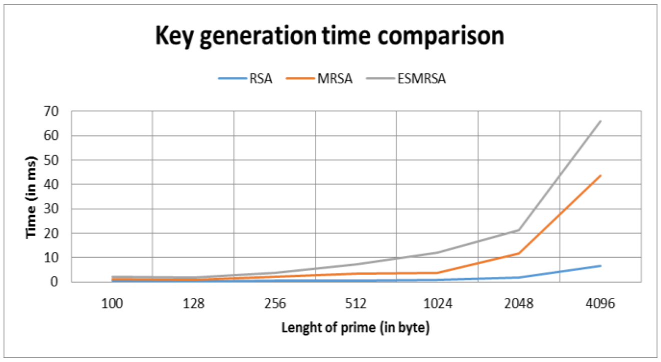


Fig. 4. Key Generation Time Comparison.

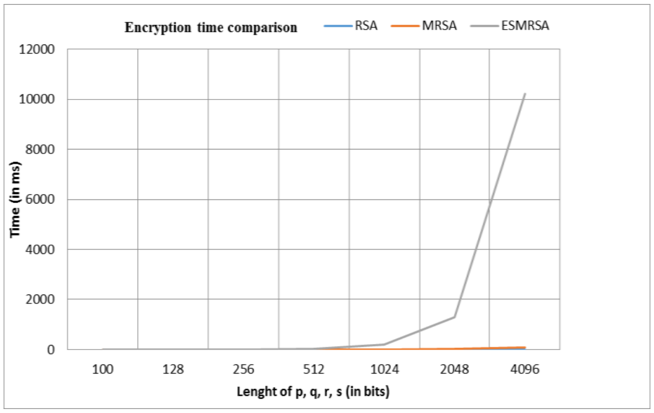


Fig. 5. Encryption Time Comparison.

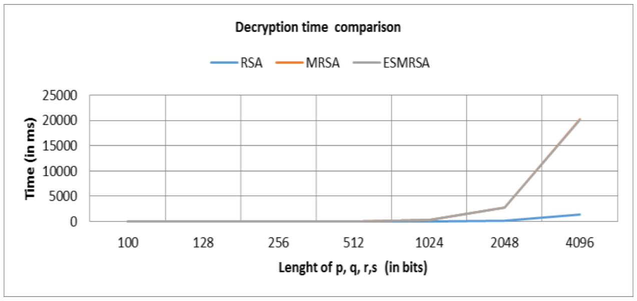


Fig. 6. Decryption Time Comparison.

C. Security Analysis

Similarly, as Muhammad explained, the factorization technique can be used to find 'w', 'x', 'y', 'z', the four prime numbers p, q, r, s that we renamed for the circumstance, but not to find 'e', 'f', 'g'. Indeed for them, you have to use the brute force attack. Thus, the simplest attack on a number is the brute force attack. In this attack, an attacker simply tries to decrypt the message with each possible secret key and checks the decryption result to see if it makes sense. With enough time and computer resources, this attack is guaranteed to work since the real secret key must be found in all possible secret keys and the attacker will eventually try it and (hopefully) realize that the resulting plain text is the right one [31].

$$\Omega_{\text{system1}} = \Omega_{w,x,y,z} + \Omega_{\text{bruteforce1}} \qquad (39)$$

For our EMSRSA

$$\Omega_{\text{system2}} = \Omega_{x,y} + \Omega_{\text{bruteforce2}} \qquad (40)$$

With RSA original

Ω_{system}= Time required to break the system.

$\Omega_{w,x,y,z}$= Time required to find w, x, y, z using, for example, the methods of modern factorization algorithms GNFS (General Number Field Sieve) and ECM (Elliptic Curve Method).

$\Omega_{x,y}$= Time required to find x, y, using, for example, the methods of modern factorization algorithms GNFS (General Number Field Sieve) and ECM (Elliptic Curve Method).

$\Omega_{bruteforce1}$= Time required for the brute force attack that would allow to find the prime numbers ad random *e*, *f* and *h*.

$\Omega_{bruteforce2}$= Time required for the brute force attack that would allow the first number e to be found at Ø(*n*).

We have just demonstrated that $\Omega_{w,x,y,z}$ is higher than $\Omega_{x,y}$.

(Key generation time)

Similarly, $\Omega_{bruteforce1}$> $\Omega_{bruteforce2}$, because it takes more time for brute force to find ad random three prime numbers (*e*, *f*, and *h*) than is required in the original RSA method.

Let's take a demonstrative example: if we use the site https://howsecureismypassword.net/, a tool for checking cryptanalysis online password, it takes at least 34,0000 years to get to the end of a password like Binl_ose1***. The longer and more complex the password, the longer it takes to crack it. We also know that there are at least (62)x possibilities to crack by brute force a key of length x. If we have the length x= 5 characters, this gives us 384400000000000000 possibilities to find in a time $\Omega_{bruteforce}$, depending on the performance of the attack server.

This confirms that $\Omega_{bruteforce1}$>$\Omega_{bruteforce2}$, because instead of working as an input with the two prime numbers taken at random, you have to find four.

We deduce that $\Omega_{system1}$>$\Omega_{system2}$. The hacker will necessarily take longer to crack our improved EMSRSA than to do it with the original RSA.

The security of our EMSRSA algorithm is therefore stronger than RSA, because it is already stronger than Muhammad's MRSA, because the hacker has to find four prime numbers instead of two, then it takes him a longer time to generate the primary key related to the factorization of the large prime number 'N'. However, our decoding time is shorter than that of MRSA from large bits. The interest of our work is that RSA is used only for the exchange of private keys, as the public key passes through the mobile medical data transmission network. For the transmission of this data, a symmetrical algorithm such as AES, for example, would be more appropriate. In this context, private key security is important, and the generation time of the primary key is very important, and this is what we have obtained by our method, by improving the security of RSA.

V. Conclusion and Future Work

The security of the transmission of our medical data depends on the encryption we apply. We chose RSA, which we modified and which uses the factorization of large numbers. Our work, similar to that of Muhammad Ariful, consisted of 'n' prime numbers instead of two. The triple encryption-decryption we have done has made it more difficult to break the factorized number 'N' on which all the encryption depends. The time required to break it in our case is much longer than that of the MRSA of Muhammad, and RSA. The encryption time is much longer than RSA and MRSA, but through our offline storage, we have been able to improve decryption time, better than Muhammad Ariful, especially when the input bits are larger. We therefore intend, in our future work, to improve the time of encryption by applying the AES algorithm at this level, which would result in an asymmetric combination of AES and EMSRSA, to increase the performance of our new cryptosystem algorithm.

References

[1] R. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, Vol.21, 1978, pp. 120–126.

[2] D. Sathya, P. G. Kumar, “Secured remote health monitoring system”, Healthcare Technology Letters, Vol. 4, Iss. 6, 2017, pp. 228–232.

[3] J. Heurix and T. Neubauer, “Privacy-preserving storage and access of medical data through pseudonymization and encryption”, Institute of Software Technology and Interactive Systems, Austria, 2011, pp. 1–12.

[4] M. Layouni, K. Verslype, M. T. Sandıkkaya, B. De Decker, and H. Vangheluwe, “Privacy-preserving telemonitoring for eHealth”, IFIP International Federation for Information Processing 2009, Data and applications security, LNCS 5645, 2009, pp. 95–110.

[5] M. Milutinovic, B. De Decker, “Privacy-preserving data management in eHealth systems”, Conference on enterprise information systems / HCIST–International Conference on Health and Social Care Information Systems and Technologies, 2013, pp. 1085–1092.

[6] Sun, H.M., M.E.,Ting, W.C. and Hinek, M.J., “Dual RSA and its security Analysis”. IEEE Transactions on information Theory,53, 2007, pp. 2922–2933.

[7] S. K. Abd and S.A.R Al-Haddad, F. Hashim and A. Abdullah, “A review of cloud security based on cryptographic mechanisms”, International Symposium on Biometrices and Security Technologies (ISBAST), 2014, pp.106–111.

[8] Ms. R. Patidar, Mrs. R. Bhartiya, “Implementation of modified RSA cryptosystem based on offline storage and prime number”, International Journal of Computing and Technology, Volume 1, Issue 2, (IJCAT), ISSN: 2348-6090, 2014, pp. 205–209.

[9] M. A. Islam, Md. As. Islam, N. Islam, B. Shabnam, “A modified and secured RSA public key cryptosystem based on ‘n’ prime numbers”, Journal of Computer and Communication, 6, 2018, pp.78–90.

[10] H. Ali, M. Al-Salami, “Timing attack prospect for RSA cryptanalysts using genetic algorithm technique” Computer science department, Zarka Private University, Jordan, The International Arab Journal of Information Technology, Vol. 1, No. 1, 2004, pp. 80–84.

[11] B. Kumar, J. Boaddh and L. Mahawar, “A hybrid security approach based on AES and RSA for cloud data”, International Journal of Advanced Technology and Engineering Exploration, Vol 3(17), 2016, pp. 43–49.

[12] A. Bhardwaja, G. Subrahmanyamb, V. Avasthic, H. Sastryd, “Security algorithms for cloud computing”, Procedia Computer Science, 85, 2016, pp.535–542.

[13] B. Swamia, R. Singhb, S. Choudharyc, “Dual modulus RSA based on Jordan-Totient function”, International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST), 2015, pp. 1581–1586.

[14] Dr. P. Mahajan and A. Sachdeva, “A study of encryption algorithms AES”, DES and RSA for Security, Global Journals Inc.(US), vol 13, 2013, pp. 8807 –8811.

[15] D. Preuveneersa, Y. Berbersa, W. Joosena , “The future of mobile e-health application development: exploring HTML5 for context-aware diabetes monitoring”, Procedia Computer Science, 21, 2013, pp. 351–359.

[16] M. Kethari, Prof. L. Desai, “A survey on secure web based medical data transmission for eHealth”, International Journal of New Innovations in Engineering and Technology, Volume 6 Issue 3, ISSN : 2319-6319, 2016, pp. 18–20.

[17] V. Kapoor and R. Yadav, “A hybrid cryptography technique for improving network security”, International Journal of Computer Applications (0975 – 8887) Volume 141 – No.11, 2016, pp. 25–30.

[18] K. G. Kadam and Prof. V. Khairnar, “Hybrid RSA-AES Encryption For Web Services”, International Journal of Technical Research and Applications e-ISSN: 2320-8163, 31, 2015, pp.51–56.

[19] K. Rege, N. Goenka, P. Bhutada, S. Mane, “Bluetooth communication using hybrid encryption algorithm based on AES and RSA", International Journal of Computer Applications (0975-8887) Volume 71-No.22, 2013, pp. 10–13.

[20] R. Raj, Y. S. Solunke, “A modified RSA cryptosystems and analysis”, Published By: Blue Eyes Intelligence Engineering & Sciences Publication Pvt. Ltd, 2015, pp. 1–3.

[21] S. Patel, P. P. Nayak, “A novel method of encryption using modified RSA algorithm and chinese remainder theorem”, Department of Electronics and Communication Engineering National Institute of Technology, 2009, pp. 1–44.

[22] A. Gupta and V. Sharma, “Modified double Mod RSA tested with brute force attack”, International Journal of Innovative Research & Development, 2014, pp. 1–4.

[23] B. Yüksel, A. K., Ö. Özkasap, “Research issues for privacy and security of electronic health services”, Future Generation Computer Systems,68, 2017, pp.1–13.

[24] S. Bhuyan, H. Kim, O. O. Isehunwa, N. Kumar, J. Bhatt, D. K. Wyant, S. Kedia, C. F. Chang, D. Dasgupta, “Privacy and security issues in mobile health: current research and future directions”, The University of Memphis School of Public Health 135 Robison Hall, 2017, pp. 188–191. https://doi.org/10.1016/j.hlpt.2017.01.004

[25] H. S. G. Pussewalage, V. A. Oleshchuk, "Privacy preserving mechanisms for enforcing security and privacy requirements in e-health solutions", International journal of Information Management,36, 2016, pp.1161–1173.

[26] Y. Li, Q. Liu, T. Li, “Design and implementation of an improved RSA algorithm”, International Conference on e-Health Networking, Digital Ecosystems and Technologies, 2010, pp. 390–393.

[27] S. Sharma, S. Hiranwal, P. Sharma, “A new variant of subset-sum cryptosystem over RSA”, International Journal of Advances in Engineering & Technology, 2012, pp. 90–97.

[28] A. A. Ayele, Dr. V. Sreenivasarao, “A modified RSA encryption technique based on multiple public keys”, International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 4, 2013, pp. 859–864.

[29] H. Huang, T. Gong, N. Ye, R. Wang and Y. Dou, “Private and secured medical data transmission and analysis for wireless sensing healthcare system”, 1551-3203, (c) IEEE, 2016, pp. 1227–1237.

[30] B. P. U. Ivy, P. Mandiwa and M. Kumar, “A modified RSA cryptosystem based on ‘n’ prime numbers”, International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume1 Issue 2, 2012, pp.63–66.

[31] https://www.commonlounge.com/discussion/4c8ace459d1840
408e487a673cca255d, accessed July 16th, at 11:36.

[32] Shambavhi, Dr. S. Sharma, “Enhanced RSA Algorithm for data security in cloud”, Iconic Research and Engineering (IRE) journals, volume1, Issue 9, 2018, pp. 64–67.

[33] A. Nag, V. K. Jain, “Implementation of Modified RSA in Matlab”, Volume 3 Issue 13, 2014, pp. 382–384.

[34] H. R. Hashim, “A New Modification of RSA Cryptosystem Based on the Number of the Private Keys”, American Scientific Research Journal for Engineering, Technology, and Sciences (ARSJETS), Volume 24, No 1, 2016, pp 270–279.

[35] E. S. I. Harba, “Secure Data Encryption Through a Combination of AES, RSA and HMAC”, Engineering, Technology & Applied Science Research Vol. 7, No. 4, 2017, pp. 1781–1785.

[36] A. H. Mansour, “Analysis of RSA Digital Signature Key Generation using Strong Prime”, International Journal of Computer (IJC), Volume 24, No 1, 2017, pp. 28–36.