International Journal for Innovation Education and Research

ONLINE ISSN: 2411-2933 PRINT - ISSN: 2411-3123

Implementation of RSA Cryptography Algorithm in Language C in Exchange of Text Messages

Ítalo Rodrigo Soares Silva;Ricardo Silva Parente;Davi Samuel Dias Maia;Paulo Oliveira Siqueira Júnior;Jorge de Almeida Brito Júnior;Manoel Henrique Reis Nascimento;Igor Felipe Oliveira Bezerra;David Barbosa de Alencar

Abstract

The encryption approach is widely used in mobile applications where there is message exchange and in banking transactions or financial pointers tools. The present research has as a proposal to approach an algorithm developed in C language and based on the RSA encryption methodology using the symmetric and asymmetric key concepts in order to demonstrate its use in the exchange of messages where there is secrecy of information, thus the importance of information security, its use has obtained interesting results with the manipulation of large scale vectors where the number of characters is a factor the process of letter inversion for encryption and decryption.

Keyword: Information Security; Data; Symmetric and Asymmetric.

Published Date: 10/31/2019 **Page.427-440** **Vol 7 No 10 2019**

DOI: https://doi.org/10.31686/ijier.Vol7.Iss10.1790

Implementation of RSA Cryptography Algorithm in Language C in Exchange of Text Messages

Ítalo Rodrigo Soares Silva
italo.computation@gmail.com
Student, Universidade Paulista UNIP - Computer Science Course - Brazil

Ricardo Silva Parente
ricardosilvaparente@gmail.com
Student, Universidade Paulista UNIP - Computer Science Course - Brazil.

Davi Samuel Dias Maia
davisdm39@gmail.com
Student, Universidade Paulista UNIP - Computer Science Course - Brazil

Paulo Oliveira Siqueira Júnior
paulojunior051996@gmail.com
Student, Universidade Paulista UNIP - Computer Science Course - Brazil

Jorge de Almeida Brito Júnior
jorge.brito@itegam.org.br
Galileo Institute of Technology and Education of the Amazon – ITEGAM - Brazil

Manoel Henrique Reis Nascimento
hreys@itegam.org.br
Galileo Institute of Technology and Education of the Amazon – ITEGAM - Brazil

Igor Felipe Oliveira Bezerra
igorfelipe.dss@hotmail.com
Galileo Institute of Technology and Education of the Amazon – ITEGAM - Brazil

David Barbosa de Alencar (Corresponding author)
david002870@hotmail.com
Galileo Institute of Technology and Education of the Amazon – ITEGAM - Brazil

Abstract

The encryption approach is widely used in mobile applications where there is message exchange and in banking transactions or financial pointers tools. The present research has as a proposal to approach an

algorithm developed in C language and based on the RSA encryption methodology using the symmetric and asymmetric key concepts in order to demonstrate its use in the exchange of messages where there is secrecy of information, thus the importance of information security, its use has obtained interesting results with the manipulation of large scale vectors where the number of characters is a factor the process of letter inversion for encryption and decryption.

Keywords: *Information Security; Data; Symmetric and Asymmetric.*

1. Introduction

The need to gain security in information is something that has been discussed for many years, when Julius Caesar emperor of Rome began with the idea of letter inversion, or in the use of algorithms and techniques in Wars to send encrypted messages in order to prevent the enemy regiment intercepts such information, soon becomes something of many times, the act of encrypting or creating models that aid in the management of information becomes necessary and allows a significant gain in the branch of Computing, understand the methods and ways studied of how to initiate such a feat is within the scope of this research. Encryption has been used for a long time and continues to be used continuously and assiduously by businesses, people and software to protect data that is passed from one place to another, based on the frequent use of encryption was implemented the software instituted by the authors of this work in order to show in a practical way the process that the algorithm does from the pre-encryption to the encryption, taking as an example the algorithm and the software development.

The existing cryptographic types are private (symmetric) keys and public keys (asymmetric), where the present research approaches methodically and technically how to enable an adaptation of the RSA algorithm built in C language to allow better understanding and because it is structural.

There is a big problem with the advancement of technology, data security, in which many people suffer from data interception, whether it is messages, images, videos, audios or other media or files.

The use of encryption becomes increasingly useful as new methods are built to address needs and failures that were once notable, the present research can demonstrate that the use of an adaptive implementation can be functional in several situations.

The main objective of this work is to implement a cryptographic algorithm based on RSA encryption methods and demonstrate its use in C language, analyzing the encryption and decryption methods and contextualizing on the concepts and principles of data security.

2 . The Importance of Data Security

Message encryption is a widely used approach in mobile applications, web systems, and financial support, as its use becomes efficient in data security, transaction of information and secrecy of content processed on the web, which are often at risk to be intercepted by several existing computational techniques.

Evidence of its importance is mentioned by Antunes and Kowada.

"[..] due to the technological evolution of mobile devices with the launch of smartphones and easy access to high-speed Internet, the limitation of the communication feature has become increasingly evident. As a

result, a number of smartphone instant messaging application initiatives have begun to emerge, similar to those currently used in personal computers" [1].

WhatsApp, Facebook, Messenger among others, both act in the connection of people and communication of data through APIs in a large network, responsible for establishing connections with different clients and performing the tasks requested by each user, in the middle of these procedures failures arise.

Montanheiro and Carvalho wrote:

"When malicious individuals identify failures, they can steal information, intercept communications between client and server, stop or even destroy the web service, leaving it unattended or unusable, causing various inconveniences to users of a service" [2].

These are obstacles such as these that cause dissatisfaction and dissatisfaction in large brands and products, the lack of security becomes a problem and ends up being irreversible when a project does not follow standards or methodologies that allow the scalability among other pillars of software engineering that concretize the quality of a product.

Montanheiro and Carvalho affirm that "Failures such as those that generate the Authentication Break and Session Management vulnerability can be avoided if they are thought of in the early phases of the project". These flaws allow attackers to steal user data from the application, either by showing details of user sessions on URLs, non-expiring sessions, or access tokens that are not invalidated when logging out. When designing and modeling a system that does not expose unnecessary data to users, this type of failure can be avoided [2].

Borges exposes the use of new technologies by criminals:

"Obviously, crime would not fail to take advantage of the opportunities offered by new technologies, and the practice of cybercrime on the Internet is perverse, with countless bank frauds, extortions from computer intrusions, viruses and programs scattered across the network to obtain data that allows criminal practice, child pornography and many other illicit or reprehensible behaviors" [3].

Then there are the data interception scanners that have automated features that after initial configuration do not require man-machine interaction, some examples are cited by Gomes, who comments on the purposes "Each scanner has its own methods and different types of attacks, the most common are" [4].

- Authentication Bypass ou Brute forcing
- SQL Injection/Blind SQL Injection
- Cross Site Scripting / Persistent Cross Site Scripting
- Command Injection!XPath Injection!SOAP/AJAX Attacks
- CSRF / HTTP Response Splitting
- Arbitrary File Upload attacks
- Remote File Include (PHP Code Injection)
- Application Errors

Soon the technologies that become viable in several aspects end up being trapped by lack of planning allowing the invasion and vulnerability in small details.

2.1 Cryptography and Description

Data encryption becomes a way to secure information crafted in a computing environment, decryption is performed in order to discover the encrypted information.

According to Zoelner et al. "A type of encryption or cipher is known as Hill Cipher, and is a type of linear algebra-based substitution cipher used for message encoding" [5].

To De Sousa Abreu "In information security jargon, cryptography provides not only to ensure the confidentiality of message content, data and information, but also to integrity and authenticity" [6].

Vicente et al. "The Emperor of Rome Julius Caesar (100-44 BC) also developed such a form of communication with his generals through the displacement of the letters of the alphabet, known today as" substitution cipher "[7].

It states that "The first existing methods such as the" Cipher of Julius Caesar "only used a coding algorithm (mono-alphabetic), therefore the sender only had to know how to subtract from the message three letters to decipher" [7].

Da Cunha comments that cryptography is a technique of maintaining secrecy about information and mainly as a means of security for communications in various technological environments where passwords are used [8].

2.2 Types of Cryptography

Encryption can be based on Symmetric and Asymmetric what for Vieria Filho and Azeredo "the cryptography can also be understood within a process of technological innovation [...] the advance of the cryptographic analysis depends on the managerial abilities that are constructed to explore and to capture knowledge" [9].

According to Silva and Oliveira:

"The need to keep secrets for man has brought the advancement of effective methods of security according to the knowledge that was within his reach [..]. With the passing of the centuries and the improvement of new techniques, security has become a very broad and researched aspect, and thus new inventions have emerged according to the necessity of the moment" [10].

Encryption consists of modifying the original text, known as plain text, into ciphertext so that it can not be read by unauthorized persons. When a text is encrypted, a key is generated that will be used by the recipient to decode the information. At first all information is public, only the key to access it is private [11].

2.3 Symmetric Cryptography

In symmetric encryption the secret key is unique, where the same key will be used to encrypt and decipher the information. To create the key, there must be an agreement between the sender and receiver of the information, since it will be used in the same algorithm before sending and receiving messages [12].

According to France:

"The problem with Symmetric Cryptographic Systems lies in the distribution of the key, which in electronically implemented methods are made through electronic channels (telephone line and radio waves), vulnerable to the" listening" of some intruder. Therefore, these must be exchanged between the parts and stored safely, which is not always possible to guarantee" [13].

This type of encryption has algorithms that perform different tasks and influence its DES, AES and IDEA

performance.

The first one determined by the Data Encryption Standard concept created by International Business Machines (IBM) in 1977, which can allow about 72 quadrillion combinations, its key size is considered small, having been broken in 1972. This was replaced by the next one that is or AES [13].

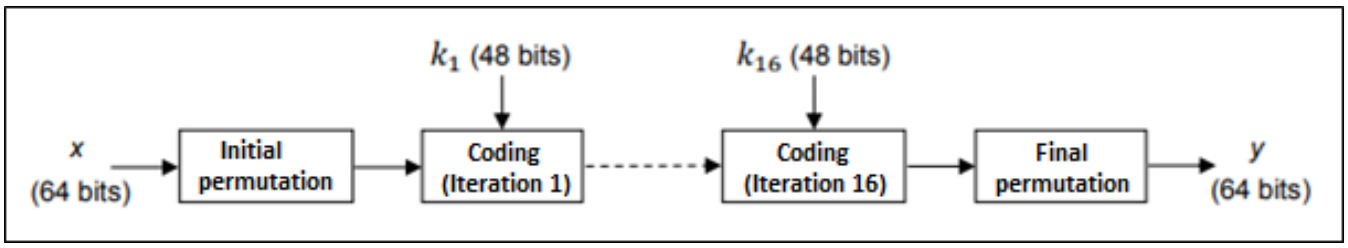

Figure - 1 Symmetric model DAES

Source: Da Silva (2018) [14].

The second concept by Advanced Encryption Standard is applied in Wi-Fi connections was created by the National Institute of Standards and Technology (NIST) in 2003, it is one of the most popular algorithms since 2006, it is fast both in software and in hardware, is relatively easy to perform and requires little memory [13].

Maia et al. explains the operation of this algorithm:

"The AES is a symmetric block cipher, where the current standard operates on a 128-bit data block, which is organized in the form of a four-order square matrix of bytes, called State, where the ordering of the bytes within the array occurs by column. Keys can be parameterized in sizes of 128, 192 and 256 bits. Each iteration, or round of encryption on each block of data (these rounds can vary according to the size of the key: 10, 12 and 14 rounds, for keys of 128, 192 and 256 bits respectively), several operations are performed: Byte substitution (SubByte), ShiftRow, MixColumns and AddRoundKey, which occurs over arithmetic in the finite body GF (28), known as the Field of Galois (GF - Galois Field), for the decryption of the data the mathematical operations are inverted)" [15].

The figure below exemplifies the model addressed by Maia et al in their research.

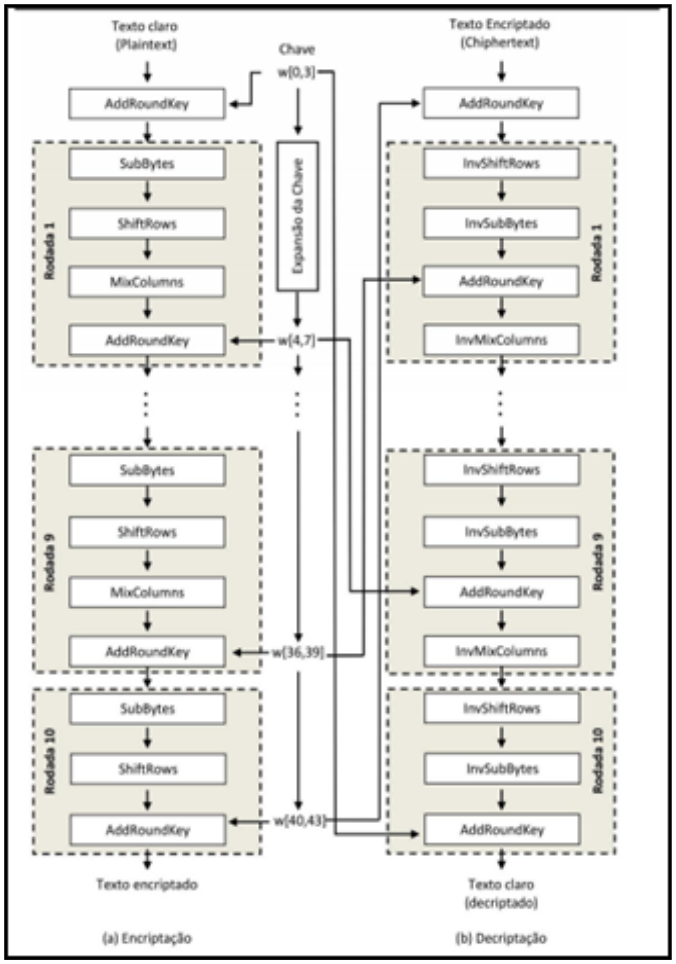

Figure - 2 AES Symmetric Model

Source: Maia et al (2017) [15].

Lastly IDEA The International Data Encryption Algorithm (IDEA) was created in 1991 by James Massey and Xuejia Lai and holds a patent from the Swiss company Ascom Systec. The IDEA is a symmetric algorithm that uses a 128-bit key, so it is considered a power in symmetric encryption.

2.4 Asymmetric Cryptography

For Braga and Dahab "Public key cryptography uses two keys that are mathematically related and constructed to work together" [16]. One of the keys is the private having the other commonly called public and they differ in the logic of the construction only by questions of operational visibility.

Pigatto comments on the functioning of the asymmetric model and its concept:

"The sender uses the public key of the receiver to encrypt open text in ciphertext and after receiving the resulting ciphertext the receiver uses its private key to decrypt the ciphertext, again retrieving readable text" [17].

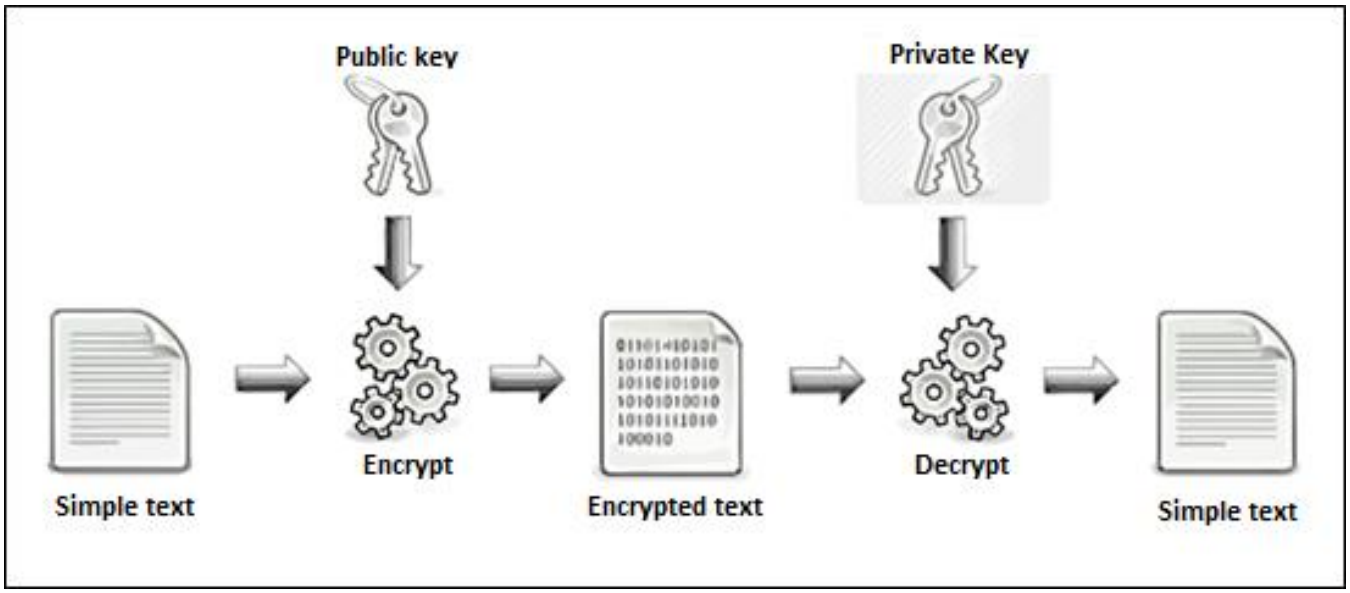

Figure - 3 Asymmetric Cryptography Scheme

Source: Pigatto (2012) [17].

In its model it is easy to determine the flow of the encryption and decryption process according to the key creation steps, where encryption occurs, but only the encryption with the key pair (private, public) can be decrypted.

2.5 Algorithm RSA

The RSA algorithm was created in 1978 by Rom Rivest, Adi Shamir and Leonard Adleman at the Massachusetts Institute of Technology (MIT) and christened with the initials of their surnames. It is currently the most widely used asymmetric cryptography method in the world, mainly in service protocols such as SSH and SSL, which manage a secure communication channel between the client and the server, which depend on the internet [18].

According to Molinari an encrypted public-key code must contain an A-coding methodology and a private decoding scheme B, where A and B are easy to calculate and for a message M, B (A (M)) = A (B (M)) = M, through this method arrive in the original message [19].

The coding key of the RSA consists essentially of n = pq, where p and q are large and distinct primes. By defining the elements, it is possible to calculate Euler's φ (x), that is, the number of numbers that are prime to each other of the chosen number. The next step is to choose a number and where 1 <e <φ (x), so that e is co-prime of φ (x). In other words, one searches for and where the MDC (φ (x) .e) = 1, where e> 1. The reader to identify the message would have to perform the inverse process to that presented, calculating to perform the decoding.

According to Filho and Azeredo the RSA algorithm is one of the most secure methods currently due to the difficulty in breaking the decoding key, that is, its complexity becomes larger when the number of elements becomes larger than is the case of the adaptation addressed in this research [20].

2.6 Proposed Algorithm

The proposed algorithm will be an adaptation of the RSA that has the best qualification in terms of the difficulty to understand cryptographic systems that act in the field of confidential data exchange or require information security, such as API's, mobile applications or financial support systems.

The implementation will be based entirely on the RSA model with small modifications, one of them will be the amount of characters used in the inversions of letters so it will be demonstrated in the methodology the vector used to compose this framework, ie a database in memory of low consumption and that allows the use of several alphanumeric characters.

According to De La Rocha Ladeira and Raugust:

The RSA holds because of the difficulty in factoring a large number (n) into prime numbers (p and q). If b is the number of bits of n, then there are √ (2b-1) possibilities to be tested in an eventual worst case, which results in time complexity of O (√ (2b)). As a matter of curiosity, considering b = 2048, √ (2b) results in a number a little larger than 1.79.10308. Considering a supermachine that can process 1 billion (109) attempts per second, it would take more than 5.10291 years [21].

Another feature to be addressed in the algorithm will be the size of the keys that will be a differential in terms of difficulty, because the larger the number generated as a key the greater the difficulty to decrypt.

2.7 The Algorithm

The RSA is strongly linked to the Theory of Numbers, being based on pillars as the rest operations and factorization by prime numbers. The algorithm can be summarized in the steps described below [22].

1. Get two prime numbers p and q;
2. Calculate n = pq;
3. Calculate Φ (n) = (p-1) (q-1);
4. Choose and | Maximum Common Divide - MDC between e and Φ (n) is equal to 1, that is, e and Φ (n) are co-primes (relative primes);
5. Calculate d | of ≡ 1 (mod Φ (n)), that is, mod Φ (n) = 1;
6. Public key: (e, n); private key: (d, n);
7. Function to encrypt a message m: C (m) = me mod n = c;
8. Function to decipher a message c: D (c) = cd mod n = m;
9. D (C (m)) = m.

3. Development

3.1 Implementation

The procedure below Fig. 4 is responsible for the choice of the numbers p, q and e, which is randomly defined, calling the primes generating function shown in Fig. -2, the variables gain prime numbers.

```
void Randomize(){
    srand((unsigned)time(NULL));
    do{
        p = rand()%150;
    }while(p == 0);

    do{
        q = rand()%150;
    }while(q == 0);
    do{
        e = rand()%150;
    }while(q == 0);

    p = nPrimo(p);
    q = nPrimo(q);
    e = nPrimo(e);
}
```

Figure - 4 Generating random numbers

Source: Authors (2019).

```
int nPrimo(int e1){
    int num = e1;
    int i, div;
    inicio:
    div = 0;
    for (i = 1; i <= num; i++) {
        if (num % i == 0) {
            div++;
        }
    }

    if(div == 2) return num;
    else{
        num++;
        goto inicio;
    }
}
```

Figure - 5 Generating cousins

Source: Authors (2019).

This structure is used to generate the value of n and to compute the function of Φ (n) = (p-1) (q-1), where símbolo is represented by the toti name of the totient function. In addition to obtaining the co-primes between e and n that will form the public key (e, n).

```
void geraChaves(){
    int primo;
    //------
    n = p*q;
    //FUNÇÃO TOTIENTE
    toti = (p-1)*(q-1);

    //CÁLCULO PARA OBTENÇÃO DE CO-PRIMOS ENTRE E e N
    //*e1 = nPrimo();
    do{
        primo = MDC(toti,e);
        e = nPrimo(e);
    }while(primo != 1);

}
```

Figure - 6 Generates keys

Source: Authors (2019).

The following is the method responsible for the maximum common divisor that is very important for the

construction of this system.

```
int MDC(int toti, int e){

    n1 = toti;
    n2 = e;
    mdc = n1%n2;
    while(mdc!=0){
        n1    = n2;
        n2    = mdc;
        mdc = n1%n2;
    }
    return n2;
}
```

Figure - 7 Maximum Common Splitter

Source: Authors (2019).

The modular function in Fig 8 is responsible for making a number represented by the distance between a point A and a point B, which in this case would be the distance between a part of the public key that is represented by the letter e and by the variable named toti, a which is used together with the variable and to create a value, this value will be assigned ad, a part of the private key that is composed of (d, n).

```
int mod(int a, int b)
{
    int r = a % b;

    /* Uma correção é necessária se r e b não forem do mesmo sinal */

    /* se r for negativo e b positivo, precisa corrigir */
    if ((r < 0) && (b > 0))
    return (b + r);

    /* Se r for positivo e b negativo, nova correção */
    if ((r > 0) && (b < 0))
    return (b + r);

    return (r);
}
```

Figure - 8 Modular function

Source: Authors (2019).

Repeated successive subtractions are what the Euclidean function does, so in programming language this Euclid method is implemented recursively until it finds a value that satisfies the necessary conditions.

```
int euclides_ext(int a, int b, int c)
{
    int r;

    r = mod(b, a);

    if (r == 0) {
    return (mod((c / a), (b / a))); // retorna (c/a) % (b/a)
    }

    return ((euclides_ext(r, a, -c) * b + c) / (mod(a, b)));
}
```

Figure - 9 Euclidean Method

Source: Authors (2019).

Function to store the encrypted text in a txt file, then this item will be read and encrypted.

```
void ARQUIVO(){
     arq = fopen("Dados.txt","w");
     int i;
     if(!arq){
        printf("\nErro ao abrir o arquivo!");
        exit(1);
     }
     fprintf(arq,"\tDados de Criptografia!!");
     fprintf(arq,"\n%i\n%i",d,n);
     fprintf(arq,"\n%i\n",strlen(frase));
     fprintf(arq,"\nVetor Convertido:");
     for(i = 0; i< strlen(frase);i++){
         fprintf(arq,"\n%i ",cripi[i]);
     }
     fclose(arq);
}
```

Figure - 10 Write file

Source: Authors (2019).

A process for reading the encrypted file and thus decrypting the message.

```
void LERARQUIVO(){
    char f[1000];
    arq = fopen("Dados.txt","r");
    int i,d1,n1,tam;
    if(arq == NULL){
        puts("\nErro ao abrir o arquivo!!");
        puts("\nFaça a criptografia de uma chave para gerar o arquivo!!\n");
    }else{
        printf("\n\n\t---LENDO O ARQUIVO-----\t\n\t\t<enter>");
        getch();
        fscanf(arq,"\n %[^\n]",&f);
        fscanf(arq,"\n%i",&d);
        fscanf(arq," %i",&n);
        fscanf(arq,"\n%i",&vTamanho);
        fscanf(arq,"\n %[^\n]",&f);
        for(i = 0; i < vTamanho;i++){
            fscanf(arq,"%i ",&cripi[i]);
        }
    }
    fclose(arq);
}
```

Figure - 11 Read file

Source: Authors (2019).

To encrypt the messages a conversion of the characters of the message into numbers is done and then the encrypted message is printed on the screen, after this step is completed the value of d is originated through the Euclidean function as already mentioned above. From there the public key is shown, where anyone can access without compromising the security of the message.

```
void criptografa(){
    int vtam = strlen(frase);
    //int cripi[vtam];
    int i;
    for(i = 0; i < vtam; i++){
        cripi[i] = exp(vFrase[i],e,n);
    }
    //----
    printf("\nFRASE CRIPTOGRAFADA:\n");
    for(i = 0; i < vtam; i++){
        printf("%i ",cripi[i]);
    }
    printf("\n");
    d = euclides_ext(e, toti, 1);
    printf("\nChave Pública: (%i,%i)",e,n);
    ARQUIVO();
}
```

Figure - 12 Encrypt

Source: Authors (2019).

The method for decrypting the encrypted message is just below and the private key is used to decipher character by character until the entire message is completely decrypted, then the message is displayed on the screen in the form of numbers, symbolizing that the encryption occurred, these numbers being the message itself.

```
void descriptografa2(int tam){
    int i;
    int descri[tam];

    for(i = 0; i < tam; i++){
        descri[i] = 0;
    }
    for(i = 0; i < tam; i++){
        descri[i] = exp(cripi[i],d,n);// % n;
    }
    printf("\n");

    int x,y;
    //--
    printf("\n\nFrase Descriptografada!!\n\n");
    for(x = 0; x < tam; x++){
        for(y = 0; y < 79; y++){
            if(descri[x] == y){
                printf("%c",ASCII[descri[x]]);
            }
        }
    }
    printf("\n\n");
}
```

Figure - 13 Encrypt

Source: Authors (2019).

4. Final Considerations

The present research was successful in analyzing the context of encryption, its importance, its use and techniques used to exchange messages, the program developed in C language was implemented through the algorithm addressed during the methodology, the results indicate that its use is made more satisfactory

when the character vector is larger for a more diverse exchange of letters and when the private and public keys are of enormous numbers such as 200 figures, small values were used for better abstraction of the reader. The RSA system can be very well accepted as a security source in mobile applications and web platforms, mainly in the use of financial support tools.

5. Acknowledgement

The Paulista University - UNIP and the Galileo Institute of Technology and Education of the Amazon - ITEGAM for technical and scientific support and collaboration.

6. References

[1] ANTUNES, Igor; KOWADA, Luis Antonio. Explorando o Sistema de Criptografia Signal no WhatsApp. In: SBSeg 2018. SBC, 2018. p. 181-195.

[2] MONTANHEIRO, Lucas Souza; CARVALHO, Ana Maria Martins. Primeiros passos para o Desenvolvimento Seguro de Aplicações Web. In: Anais Estendidos do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. SBC, 2018. p. 233-242.

[3] BORGES, Fabiani.Terrorismo Cibernético e a Proteção de Dados Pessoais. Escola Paulista de Direito - EPD. p. 1 – 3. 2015.

[4] GOMES, Eduardo H. Segurança em sistemas de e-learning: uma análise do ambiente Tidia-Ae/Sakai. Perspectiv@ s, v. 14, n. 13, p. 18-24, 2019.

[5] ZOELNER, Éverton Gabriel et al. CRIPTOGRAFIA. Anais do EVINCI-UniBrasil, v. 3, n. 1, p. 333-333, 2018.

[6] DE SOUZA ABREU, Jacqueline. Passado, presente e futuro da criptografia forte: desenvolvimento tecnológico e regulação. Revista Brasileira de Políticas Públicas, v. 7, n. 3, p. 24-42, 2018.

[7] VICENTE, Aparecido et al. A CRIPTOGRAFIA E SUA IMPORTÂNCIA NA ATUALIDADE. Aten@-Revista Digital de Gestão & Negócios-, v. 1, n. 1, 2016.

[8] DA CUNHA, Rhuan Gonzaga et al. CRIPTOGRAFIA DE DADOS UTILIZANDO MATRIZES. RE3C-Revista Eletrônica Científica de Ciência da Computação, v. 11, n. 1, 2016.

[9] VIEIRA FILHO, José Eustáquio Ribeiro; AZEREDO, Paula Prestes. Tecnologia, criptografia e matemática: da troca de mensagens ao suporte em transações econômicas. Desenvolvimento Socioeconômico em Debate, v. 2, n. 2, p. 22-31, 2017.

[10] SILVA, Mariana Godoy; OLIVEIRA, Cintia Carvalho. O USO DA CRIPTOGRAFIA EM ÁUDIO. Anais do Seminário de Pesquisa e Inovação Tecnológica-SEPIT, v. 1, n. 1, 2017.

[11] TANENBAUM, Andrew S. Redes de Computadores. 5º edição. Rio de Janeiro: Campus, 2011.

[12] PETRI, Marcelo. Esteganografia. Sociedade Educacional De Santa Catarina – SOCIESC, Instituto Superior Tupy. Joinville, TCC de sistemas de informação. 2004.

[13] FRANÇA, Waldizar Borges de Araújo. A utilização da criptografia para uma aprendizagem contextualizada e significativa. p. 36-37. 2016.

[14] DA SILVA, Guilherme Gomes Felix. Formalização de Algoritmos de Criptografia em um Assistente de Provas Interativo. 2018. Tese de Doutorado. PUC-Rio.

[15] MAIA, William Pedrosa et al. Projeto, implementação e desempenho dos algoritmos criptográficos AES, PRESENT e CLEFIA em FPGA. 2017.

[16] BRAGA, Alexandre; DAHAB, Ricardo. Introdução à Criptografia para Programadores: Evitando Maus Usos da Criptografia em Sistemas de Software. Caderno de minicursos do XV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais-SBSeg 2015, p. 1-50, 2015.

[17] PIGATTO, Daniel Fernando. Segurança em sistemas embarcados críticos-utilização de criptografia para comunicação segura. 2012. Tese de Doutorado. Universidade de São Paulo.

[18] STALLINGS, W. Criptografia e segurança de redes: princípios e práticas. Pearson Prentice Hall. Protocolo Diffie-Hellman Sobre Curvas Elípticas. 2008.

[19] MOLINARI, José Robyson Aggio et al. Números Primos e a Criptografia RSA. 2016.

[20] FILHO, José Eustáquio Ribeiro Vieira; AZEREDO, Paula Prestes. Tecnologia, criptografia e matemática: da troca de mensagens ao suporte em transações econômicas. Revista Desenvolvimento Socioeconômico em debate v, v. 2, n. 2, p. 23, 2016.

[21] DE LA ROCHA LADEIRA, Ricardo; RAUGUST, Anderson Schwede. Uma análise da complexidade do algoritmo RSA implementado com o teste probabilístico de Miller-Rabin. Revista de Empreendedorismo, Inovação e Tecnologia, v. 4, n. 1, p. 24-33, 2017.

[22] Rivest, R. L., Shamir, A., & Adleman, L. M. (1978, Fev.). A Method for Obtaining Digital Signatures and Publk-Key Cryptosystems. Communications of the ACM, 21(2), 120-126.