

DEVELOPING AN APPLICATION OF RSA ALGORITHM WITH JAVA

M. Nusret SARISAKAL¹ Selcuk SEVGEN² Dogal ACAR³

^{1,2,3} Istanbul University, Faculty of Engineering, Department of Computer Engineering,
34850, Avcilar, Istanbul, Turkey

¹e-mail: nsarisakal@istanbul.edu.tr ²e-mail: sevgens@istanbul.edu.tr ³e-mail: dogal@istanbul.edu.tr

Keywords: Security, Cryptography, JAVA, RSA Algorithm

ABSTRACT

This paper focuses on public-key encrypting systems that can be used for providing security on internet. It explains the RSA algorithm step by step and a Java application that encrypts and decrypts data with RSA algorithm that will be used for internet applications as a Java applet.

definitions, the mathematical model of encrypting is as follows:

$$F(M)=S \quad (1)$$

Inverse encrypting function for decrypting is as follows:

$$S(M)=F \quad (2)$$

1. INTRODUCTION

Today, by becoming widespread of Internet, the need of security on digital environment has appeared. Various decrypting and encrypting algorithms has been offered to transmit data in a secure environment. In this study, we will explain Public-key encrypting systems, RSA algorithm and Java programming language. In fact, we will explain a RSA application we have developed with Java.

Rivest, Shamir and Adleman have realized first application of public-key crypto and signing systems at 1978. This application is called RSA. RSA is based on the problem of factorization of big integers which is an another hard mathematic problem. Developing this algorithm caused more effective methods for factoring. A lot of developments occurred in 1980s, but none of them made RSA a distrustful algorithm [1].

Explaining of the terms that are used in cryptography literature respectively are as follows: It is defined that sender is a person sending a message, receiver is a person receiving a message. Encrypting is to change a message using nonlinear functions. Inverse encrypting is to decrypt a encrypted message, that is to say decrypting code using inverse of mathematical function.

Show that M is the plaintext, S is the encrypted text and F is encrypt function. It can be that the length of the encrypted text is more than the length of the plaintext. In this case, we try to shorten the encrypted text by compress algorithms. In respect of these

Generally, it is obvious that encrypting and decrypting algorithms are inverse functions each other. But this functions are nonlinear so this is making difficult the encryption and decryption processes[2].

2. PUBLIC-KEY ENCRYPTING SYSTEMS

One of the most important opinions about public-key encrypting systems is that this systems can never provide unconditional security. Because other side can find value x obtaining $y=e_k(x)$, by using e_k rule by using encrypted expression y. x expression is decrypted answer of expression y. Therefore, studies are made on computable security of public-key encrypting algorithm.

It is preferred that suitable public-key algorithm is easily computable and invers transforming function is also hardly computable. This is generally called *injectivity*.

As a result, e_k is preferred to be an injective function. Injective functions have important role on the encrypting. It is used this kind of functions to develop public-key encrypting systems.

If we talk about public-key encrypting systems, It is not enough that we have an injective function. If e_k continues to be injective function for authorized persons, It can not be possible that they decrypt the codes. The owner of the text must be able to reach the original text easily, by eliminating the injective function. This causes the necessity that the owner of

the text must be able to find the inverse of e_k rule through an extra information that she/he knows, by using a kind of secret passage.

Injective functions are used in the RSA systems. But authorized person can obtain easily the rule of inverse transforming through the information called private-key. It is useful to explain two subjects about modular arithmetic and numbers theory, before talking about details of RSA.

3. EUCLIDEAN ALGORIHM

$b \in Z_n$ has a multiplicative inverse if and only if $\text{GCD}(b,n)=1$ (Greatest Common Divisor) and that the number positive integers less than n and relatively prime to n is $\Phi(n)$ [1].

The set residues modulo n that are relatively prime to n is denoted as Z_n^* . Multiplacation modula n is associative and commutative and 1 is the multiplicative identity. Any element in Z_n^* will have a multiplicative inverse (which is also in Z_n^*).

Finally , Z_n^* is closed under multiplication since xy is relatively prime to n whenever x and y are relatively prime to n . At this point we know that any $b \in Z_n^*$ has a multiplicative inverse, b^{-1} . We can use the Euclidean Algorithm which is developed to compute this value effectively.

First, we describe the Euclidean Algorithm, in its basic form, which is used to compute the greatest common divisor of two positive integers, say r_0 and r_1 , where $r_0 > r_1$. The Euclidean Algorithm consists of performing the following sequence of divisions:

$$\begin{aligned} r_0 &= q_1 r_1 + r_2, & 0 < r_2 < r_1 \\ r_1 &= q_2 r_2 + r_3, & 0 < r_3 < r_2 \\ &\dots & \\ r_{m-2} &= q_{m-1} r_{m-1} + r_m, & 0 < r_m < r_{m-1} \\ r_{m-1} &= q_m r_m \end{aligned}$$

then it is not hard to show that

$$\text{GCD}(r_0, r_1) = \text{GCD}(r_1, r_2) = \dots = \text{GCD}(r_{m-1}, r_m) = r_m$$

Hence, it follows that $\text{GCD}(r_0, r_1) = r_m$. Since the Euclidean Algorithm computes greatest common divisors, it can be used to determine if a possible integer $b < n$ has a multiplicative inverse modulo n , by starting with $r_0 = n$ and $r_1 = b$. However, it does not compute the value b the multiplicative inverse.

Now, suppose we define a sequence of numbers to t_0, t_1, \dots, t_m according to the following recurrence (where q_j 's are defined as above).

$$\begin{aligned} t_0 &= 0 & t_1 &= 1 \\ t_j &= t_{j-2} - q_{j-1} t_{j-1} \pmod{r_0} & & \text{if } j \geq 2. \end{aligned}$$

Then, we have following useful results.

Theorem: If t_j and r_j defined like Euclidean Algorithm and t_j is defined like above recurrence, we obtain $r_j \equiv t_j r_1 \pmod{r_0}$ for $0 \leq j \leq m$ [1].

Proof: We will try to prove by induction on j . Our assertion is true for $j=0$ and $j=1$. Assume the assertion is true for $j=i-1$ and $j=i-2$ values, where $i \geq 2$. Prove that our claim is true for $j=i$. By induction, we have that

$$\begin{aligned} r_{i-2} &\equiv t_{i-2} r_1 \pmod{r_0} \\ r_{i-1} &\equiv t_{i-1} r_1 \pmod{r_0} \end{aligned}$$

now we compute :

$$\begin{aligned} r_i &= r_{i-2} - q_{i-1} r_{i-1} \\ &\equiv t_{i-2} r_1 - q_{i-1} t_{i-1} r_1 \pmod{r_0} \\ &\equiv (t_{i-2} - q_{i-1} t_{i-1}) r_1 \pmod{r_0} \\ &\equiv t_i r_1 \pmod{r_0} \end{aligned}$$

Thus we have proved the theorem through induction method.

4. CHINESE REMAINDER THEOREM

Chinese Remainder Theorem is really a method of certain systems of congruences. Suppose m_1, \dots, m_r are pairwise relatively prime (that is $\text{GCD}(m_i, m_j) = 1$ if $i \neq j$). Suppose a_1, \dots, a_r are integers, and consider the following system of congruence:

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\dots \\ x &\equiv a_r \pmod{m_r} \end{aligned}$$

Chinese Remainder Theorem asserts that this system has unique solution modulo $M = m_1 * m_2 * \dots * m_r$. However, an effective algorithm will be given for solution of this type of congruences systems. At this step, the function definition has to be investigated as follows:[1]

$$\begin{aligned} \pi : Z_m &\rightarrow Z_{m_1} \times \dots \times Z_{m_r} \\ \pi(x) &= (x \pmod{m_1}, \dots, x \pmod{m_r}) \end{aligned}$$

Example: Suppose $r=2$, $m_1=5$ and $m_2=3$, so $M=15$. Function π has the following values.

$\pi(0) = (0, 0)$	$\pi(1) = (1, 1)$	$\pi(2) = (2, 2)$
$\pi(3) = (3, 0)$	$\pi(4) = (4, 1)$	$\pi(5) = (0, 2)$
$\pi(6) = (1, 0)$	$\pi(7) = (2, 1)$	$\pi(8) = (3, 2)$
$\pi(9) = (4, 0)$	$\pi(10) = (0, 1)$	$\pi(11) = (1, 2)$
$\pi(12) = (2, 0)$	$\pi(13) = (3, 1)$	$\pi(14) = (4, 2)$

Computing Chinese Remainder Theorem is based proving that function π is a bijection. This clearly seen in above example. In fact, we can give general function for the inverse function π^{-1} .

For $1 \leq i \leq r$, define

$$M_i = \frac{M}{m_i}$$

then see $\text{GCD}(M_i, m_i) = 1$

$y_i = M_i^{-1}$ for $1 \leq i \leq r$ then obtain $M_i y_i \equiv 1 \pmod{m_i}$
For $1 \leq i \leq r$, define a function $p: Z_{m_1} \times \dots \times Z_{m_r} \rightarrow Z_m$ as follows:

$$p(a_1, \dots, a_r) = \sum_{i=1}^r a_i M_i y_i \pmod{M}$$

For example, If we solve function $p = \Pi^{-1}$, we see that it produces a clear function to solve the appropriateness of original system.

Suppose $X = p(a_1, \dots, a_r)$. if we accept $1 \leq j \leq r$ then $a_i M_i y_i$ in the above example

If $i = j$, $a_i M_i y_i \equiv a_i \pmod{m_i}$

Then we can write $M_i y_i \equiv 0 \pmod{m_i}$

Other side, If $i \neq j$ then $a_i M_i y_i \equiv 0 \pmod{m_j}$

since $m_j \mid M_i$ in this case. We have that

$$X \equiv \sum_{i=1}^r a_i M_i y_i \pmod{m_j}$$

$$X \equiv a_j \pmod{m_j}$$

When this result is true for all $1 \leq j \leq r$, it is a reliable solution for X systems appropriateness. At this point, it is necessary that we show unique solution of X is mod M . It is possible to do by a simple counting. Function Π is a defined function from a domain of cardinality M to a range of cardinality M . We have proved that function Π is a surjective function. But at the same time function Π must be an injective function (one-to-one). If so, Π is a bijection and $\Pi^{-1} = p$. Π^{-1} is a linear function for its arguments a_1, \dots, a_r .

Example: Suppose $r=3$, $m_1=7$, $m_2=11$ and $m_3=13$ then $M=1001$. Compute $m_1=143$, $m_2=91$, $m_3=77$. Compute also $y_1=5$, $y_2=4$, $y_3=12$ subject to these values [1].

$$\pi^{-1}: Z_7 \times Z_{11} \times Z_{13} \rightarrow Z_{1001}$$

$$\pi^{-1}(a_1, a_2, a_3) = 715a_1 + 364a_2 + 924a_3 \pmod{1001}$$

For example, if $x \equiv 5 \pmod{7}$, $x \equiv 3 \pmod{11}$ and $x \equiv 10 \pmod{13}$

Then this formula tells us that

$$x = 715 \times 5 + 364 \times 3 + 924 \times 10 \pmod{1001}$$

$$= 13907 \pmod{1001}$$

$$= 894 \pmod{1001}$$

5. RSA ALGORITHM

Let p and q be prime numbers, $n=pq$, $P=C=Z_n$, and define

$$K = \{(n, p, q, a, b) : n = pq \text{ } p \text{ and } q \text{ are primes,}$$

$$ab \equiv 1 \pmod{\phi(n)}\}$$

for $K=(n,p,q,a,b)$ define

$$e_k(x) = x^a \pmod{n} \text{ and}$$

$$d_k(y) = y^a \pmod{n} \text{ (} x, y \in Z_n \text{)}$$

n and b are public, p, q, a are secret.

RSA crypto system is defined computations in Z_n , where n is the product of two distinct odd primes p and q . This is $\Phi(n)=(p-1)(q-1)$ for value n . We have given formal definition of RSA crypto system above. Now, Let's verify that encrypting and decrypting are inverse operations. Since,

$$ab \equiv 1 \pmod{\Phi(n)}$$

we have that

$$ab = t\Phi(n) + 1$$

for some integers $t \geq 1$

suppose that $x \in Z_n^*$; then we have

$$(x^b)^a \equiv x^{t\Phi(n)+1} \pmod{n}$$

$$\equiv (x^{\Phi(n)})^t x \pmod{n}$$

$$\equiv 1^t x \pmod{n}$$

$$\equiv x \pmod{n}$$

6. AN APPLICATION WHICH USES RSA ALGORITHM

We will see usage of RSA algorithm through the sample application we have developed. Our program was designed that it includes necessary operations to realize operations of encryption and decryption. Java programming language was used to develop this application.

6.1. JAVA PROGRAMMING LANGUAGE

Java is an object oriented programming language, developed by Sun Microsystems at 1991. Java has class library which includes basic data types, input/output functions and other functions. This programming language also has functions which support most internet protocols. Although Java doesn't include many instructions, it is accepted as a power language, since it includes reliable instructions [4].

Although Java was not designed as a programming language of Internet, it has succeeded in Internet language in a short time. The most advantage of Java is that developed programs can run on many digital devices, that is to say being independent of platform. This devices are : Computers, machine tools, automobiles,

power counters, thief alarms and other home tools, etc. [5].

Popularity of Java has increased since it provides to do many things on the internet. But it is wrong to think Java as a language which is used for designing a web page. The independent of platform is the most important attribute of developing philosophy of Java. A software written on any platform with Java can be used on another platform without changing anything [6].

6.2. DEVELOPED APPLICATION

RSA Algorithm uses two key values. Any encrypting or decrypting operation needs the key values. Key values which are used to run the program at the moment can be computed by the program or we can run the program for key values which are determined before.

After determination of the key values, we load the file which will be encrypted, through the "load file" option in the File Menu. The same operation will be done for encrypted file at the decrypting operation.

After we have produced key values and read the file which will be encrypted, we can realize encrypting operation. We encrypt the file choosing the encrypt option in the Operation Menu. Our program produces two files after encrypting operation. One of them, chipper.txt, includes encrypted expression. chipper_int.txt consists integer equivalent of produced values with RSA algorithm after process of encrypting.

Example: Following integer values will be produced for plaintext "Deneme İfadesi" (Figure 1).

6725,14880,3028,14880,6820,14880,0,6295,8928,152
97,4648,14880,4916,6295,784



Figure 1

We can return back the original information by using chipper.txt which includes encrypted information.

7. CONCLUSION AND FURTHER PLANS

Application here was realized encrypting and decrypting by RSA Algorithm which is in the literature. While we were developing the application, we have chosen Java which is the independent of platform. This is a pre-study for future internet applications. Since RSA is a slow algorithm, it is appropriate to encrypt short messages mostly. It can be used to encrypt secret keys which are transmitted between sides in the encrypting systems using secret keys.

Evaluation of this study still continues. It will be used as a Java applet for internet applications. After it finishes, we will test and compare other encrypting algorithms.

REFERENCES

1. Stinson D. R., Cryptography Theory and Practice, CRC Press, 1995, Florida
2. William S., Network And Internetwork Security Principles And Practice, Prentice-Hall, Inc. 1995, New Jersey.
3. SARISAKAL M. Nusret, KARAHOCA Adem, DES Algoritmasını Kullanan Güvenilir Bir E-Posta İletim Uygulaması: Tuğra, I.U. Engineering Faculty, Journal of Electrical & Electronics Vol. 1, No. 1, pp 23-31, 2001.
4. <http://www.sun.com/java>
5. Chorafas D. N, Java – A Contarariann View 157-179, Visual Programming Technology, McGraw-Hill, New York, ISBN 0-07-011685-7, 1997.
6. SARISAKAL M. Nusret, UYSAL Mithat, Web Teknolojilerindeki Hızlı Gelişmelerin ve Web Programlama Araçlarının İncelenmesi, I.U. Engineering Faculty, Journal of Electrical & Electronics Vol. 1, No.1, pp 6-16, 2001.