



CRIPTOGRAFIA RSA: UMA ABORDAGEM COMPUTACIONAL

Polion Barboza de Souza e Silva Pereira

Trabalho de Conclusão do Curso Superior de Licenciatura em Matemática, orientado pela Prof^ª. Ma. Gabriela Cotrim de Moraes

IFSP
São Paulo
2018

**IFSP - Instituto Federal de Educação, Ciência e Tecnologia de
São Paulo**

CRIPTOGRAFIA RSA: UMA ABORDAGEM COMPUTACIONAL

Polion Barboza de Souza e Silva Pereira

Monografia apresentada ao Instituto Federal de Educação, Ciência Tecnologia - São Paulo, orientado pela Prof^a. Ma. Gabriela Cotrim de Moraes, em cumprimento ao requisito para obtenção do grau acadêmico de Licenciatura em Matemática.

IFSP
São Paulo
2018

Catálogo na fonte
Biblioteca Francisco Montojo - IFSP Campus São Paulo
Dados fornecidos pelo(a) autor(a)

P436c Pereira, Polion Barboza de Souza E
 Criptografia rsa: uma abordagem computacional
 / Polion Barboza de Souza E Pereira. São Paulo:
 [s.n.], 2018.
 38 f. il.

Orientadora: Gabriela Cotrim de Moraes

Trabalho de Conclusão de Curso (Licenciatura
em Matemática) - Instituto Federal de Educação,
Ciência e Tecnologia de São Paulo, IFSP, 2018.

1. Teoria dos Números. 2. Números Primos . 3.
Criptografia Rsa. I. Instituto Federal de
Educação, Ciência e Tecnologia de São Paulo II.
Título.

CDD 510

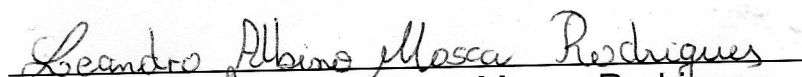
POLION BARBOZA DE SOUZA E SILVA PEREIRA

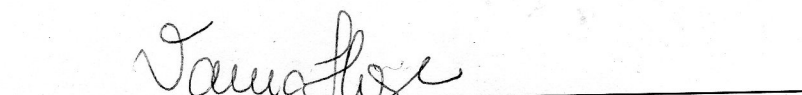
CRIPTOGRAFIA RSA: UMA ABORDAGEM COMPUTACIONAL


Monografia apresentada ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, em cumprimento ao requisito exigido para a obtenção do grau acadêmico de Licenciada em Matemática.

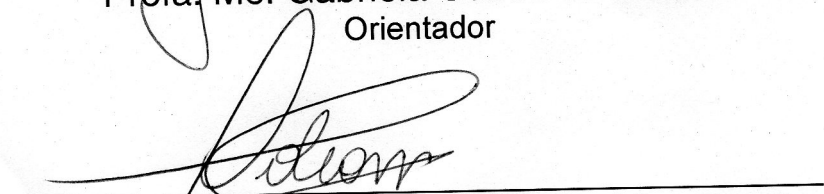
APROVADO EM 05/07/2018

CONCEITO: 7,5


Prof. Me. Leandro Albino Mosca Rodrigues
Membro da Banca


Profa. Me. Vania Batista Flose Jardim
Membro da Banca


Profa. Me. Gabriela Cotrim de Moraes
Orientador


Aluno: Polion Barboza de Souza e Silva Pereira

"A matemática é o alfabeto com qual Deus escreveu o universo"
Galileu Galilei

"Dedico este trabalho aos meus pais e a todos
os professores e colegas que trabalham para
melhoria de nossa educação"

Agradecimentos

Agradeço primeiramente a Deus e aos meus pais, que possibilitaram a oportunidade de me dedicar ao estudo e ao meu desenvolvimento como pessoa e cidadão. Também a todos os professores que foram de grande valia para meu crescimento intelectual e profissional.

Agradeço a minha orientadora Prof^a. Ma. Gabriela Cotrim de Moraes, pois juntos tivemos a oportunidade de construir este trabalho de conclusão de curso. Aos meus amigos Débora e Luiz Fernando que sempre estiveram comigo durante o período de minha graduação. Em especial ao meu amigo e colega de disciplinas Douglas, por sempre estar ao meu lado nos estudos para diversas provas e trabalhos que tivemos.

Resumo

Este trabalho de conclusão de curso apresenta um estudo da teoria a respeito das funcionalidades da criptografia RSA e o desenvolvimento de um programa de codificação e decodificação de mensagens aplicando este método. São abordados conceitos importantes para o desenvolvimento do método de criptografia RSA, um breve relato histórico da criptografia, a descrição deste método e o processo de criação de um algoritmo computacional que utilizou a linguagem *Java* para sua implementação.

Palavras - chave: Teoria dos Números, Números Primos, Criptografia RSA.

Abstract

This undergraduate thesis contains a study of the theory behind of RSA encryption functionalities and the development of a program witch encodes and decodes messages explaining the functionality of this method. There are handled important concepts for the development of the methods of RSA encryption, a brief historical report of encryption, the description of this method and the process of creation of a computational algorithm used *Java* for it's implementation .

Word - Key: Number Theory, Prime Numbers, RSA Encryption.

Lista de Figuras

1	Tela Inicial do Programa RSA	30
2	Processo de codificação	31
3	Processo de decodificação	32

Lista de Tabelas

1	Cálculo do Algoritmo Euclidiano	16
2	Cálculo do $mdc(372, 161)$	16
3	Conversão de letras em números	25
4	Blocos Codificados	27
5	Blocos Decodificados	27
6	Tabela ASCII	29

Sumário

Introdução	11
1 Conceitos Matemáticos Preliminares	13
1.1 Definições	13
1.2 Algoritmos	14
1.3 Propriedades de Congruência	16
1.4 Teoremas	19
2 Criptografia	23
2.1 Método de Criptografia RSA	24
3 Implantação da Criptografia RSA	25
3.1 Pré Codificação	25
3.2 Codificação e Decodificação	26
3.3 Descrição matemática do método	28
3.4 Segurança do sistema RSA	28
4 Programa de Criptografia RSA	29
4.1 Código Java do Processo de Criptografia RSA	32
5 Considerações Finais	35
Referências	37

Introdução

A motivação inicial para o desenvolvimento deste trabalho de pesquisa ocorreu ao estudar as disciplinas Teorias dos Números e Interface da Matemática com a Informática, durante o quarto e quinto semestres do curso de Licenciatura em Matemática.

A necessidade de desenvolver os conhecimentos com relação aos números primos e a sua importância para atividades no mundo atual, como por exemplo, a segurança no campo da informatização desencadeou a busca pelo estudo do sistema de criptografia RSA e seu funcionamento, que pode ser entendido ao estudarmos as teorias que envolvem os números primos.

A teoria na qual se apoia a criptografia RSA é a área da Matemática conhecido como Teoria dos Números. Algumas propriedades dos números inteiros como a existência da decomposição em fatores primos, relação entre pares de números inteiros como congruência e máximo divisor comum são fundamentais para a criptografia RSA.

Segundo Freitas, Souza e Agustini (2004), com os avanços tecnológicos e o processo de informatização, houve uma grande preocupação com a necessidade de proteção das informações pessoais e empresariais. A criptografia passou, então, a ser uma resposta para tal necessidade como uma forma de modificar mensagens durante o processo de transmissão, de modo que somente o receptor autorizado pudesse ler a mensagem enviada.

O método de criptografia RSA (Rivest Shamir Adleman), nome dado com base nos nomes de seus criadores, é o mais conhecido, e será estudado neste trabalho de conclusão de curso, cujos objetivos são o estudo da teoria por trás das funcionalidades da criptografia RSA e o desenvolvimento de um programa de codificação e decodificação de mensagens mostrando a funcionalidade desse método.

Segundo Coutinho (2009), os principais problemas para a implementação da criptografia RSA são: como fatorar um número inteiro de maneira eficiente e como determinar se um dado número inteiro é primo. Para isso é necessário o conhecimento das propriedades dos números inteiros.

No desenvolvimento deste trabalho, a metodologia utilizada foi o levantamento bibliográfico de autores que já haviam desenvolvido estudos sobre teoria dos números e criptografia RSA. Será também desenvolvido um programa computacional de codificação e decodificação de mensagens a partir do método de criptografia estudado.

Este trabalho de conclusão de curso está dividido em cinco capítulos. O primeiro capítulo aborda os conceitos importantes para o desenvolvimento do método de criptografia RSA, o segundo capítulo, traz um levantamento histórico da criptografia e do método RSA em particular, o terceiro capítulo apresenta a descrição do método de criptografia RSA, o quarto capítulo mostra o processo de criação de um algoritmo computacional de criptografia RSA e por fim, o quinto capítulo traz as considerações finais relevantes a este trabalho.

1 Conceitos Matemáticos Preliminares

O método de criptografia RSA tem como requisito o conhecimento de alguns algoritmos e definições importantes de Teoria dos números, necessários ao funcionamento do programa.

1.1 Definições

Consideramos as definições a seguir para este trabalho de pesquisa.

Definição 1 Seja $p \in \mathbb{Z}$ com $p \neq \pm 1, 0$, dizemos que p é um **número primo**, se e somente se, seus únicos divisores são ± 1 e $\pm p$.

Definição 2 Dados dois inteiros a e d , dizemos que d **divide** a , se existe $q \in \mathbb{Z}$, tal que $a = dq$. E denotamos por $d|a$.

Definição 3 Sejam dois inteiros a e b . Um número inteiro d será dito um **divisor comum** de a e b , se $d|a$ e $d|b$. Denotamos por $\mathbf{D}(a, b)$ o conjunto dos divisores comuns de a e b .

Definição 4 Dizemos que um número inteiro $d \geq 0$ é o **máximo divisor comum** de a e b e denotamos por $\mathbf{mdc}(a, b)$, se possuir as seguintes propriedades:

- (i) d é um divisor comum de a e b ; e
 - (ii) d é divisível por todo divisor comum de a e b .
- Se $\mathbf{mdc}(a, b) = 1$, dizemos que a e b são primos entre si.

Definição 5 Dado conjunto X , uma relação R entre pares de elementos de X é dita uma **relação de equivalência** sobre X , quando são satisfeitas as propriedades a seguir:

- (i) Para todo $x \in X$, $x R x$; (reflexiva)
- (ii) Dados x e $y \in X$, $x R y$, então $y R x$; (simétrica)
- (iii) E dados $x, y, z \in X$, $x R y$ e $y R z$, então $x R z$. (transitiva)

Definição 6 Dados dois inteiros a e b , diz-se que a e b são **congruentes módulo n** , se $n|a - b$. Denotamos congruência módulo n por $\mathbf{a} \equiv \mathbf{b}(\bmod n)$.

Definição 7 Chamamos de **sistema completo de resíduos módulo n** a todo conjunto de números inteiros cujos restos da divisão por n são os números $0, 1, 2, \dots, n - 2, n - 1$, sem repetições e numa ordem qualquer.

Definição 8 Um conjunto \mathbf{S} de números inteiros diz-se um **ideal** de \mathbb{Z} se:

- (i) Dados α e $\beta \in \mathbf{S}$, $\alpha + \beta \in \mathbf{S}$;
- (ii) Dados $\alpha \in \mathbf{S}$ e $a \in \mathbb{Z}$, $\alpha a \in \mathbf{S}$.

Definição 9 Para cada inteiro $n \geq 1$, indicaremos por $\phi(n)$ o número de inteiros positivos menores ou iguais a n , que são primos com n e chamamos **Função ϕ de Euler**.

Axioma 1 (Princípio da Boa Ordem) Considere o Conjunto $\mathbb{S} \neq \emptyset$ e formado por inteiros, existe m tal que m é **mínimo** de \mathbb{S} e denotamos por $\mathbf{m} = \min \mathbb{S}$

1.2 Algoritmos

Os algoritmos que demonstraremos a seguir servem de base e garantem a funcionalidade da criptografia RSA.

Algoritmo 1 (Algoritmo da Divisão de Inteiros) Sejam a e b inteiros, com $b \neq 0$, existem números inteiros q e r tais que: $a = bq + r$ e $0 \leq r < |b|$. Além disso, os valores de q e r satisfazendo $a = bq + r$, são únicos.

Demonstração

Começaremos mostrando a existência de q e r .

Considere $b > 0$ e $a \geq 0$. Tomamos o conjunto \mathbf{S} assim definido, $\mathbf{S} = \{a - bx, x \in \mathbb{Z} | a - bx \geq 0\}$

Se $x = 0$, temos $a - b \cdot 0 = a$, então $a \in \mathbf{S}$, logo $\mathbf{S} \neq \emptyset$.

Como $\mathbf{S} \neq \emptyset$ e formado por elementos inteiros, pelo Princípio da Boa Ordem, existe $r = \min \mathbf{S}$ e como $r \in \mathbf{S}$, então existe $q \in \mathbb{Z}$, tal que:

$$r = a - b \cdot q \iff a = b \cdot q + r$$

Para mostrar que $0 \leq r < b$ pois $b > 0$, vamos supor $r \geq b$. Nesse caso,

$$a - b(q + 1) = a - bq - b = r - b \geq 0$$

Assim $r - b \in \mathbf{S}$, o que é absurdo, pois $r = \min \mathbf{S}$. Portanto $r < b$ e como $r \in \mathbf{S}$ então $r \geq 0$.

Se $b > 0$ e $a < 0$, existem q' e $r' \in \mathbb{Z}$, tais que: $|a| = bq' + r'$ e $0 \leq r' < b$.

Se $r' = 0$, temos $-|a| = a = b(-q') + 0$, e o par $q = -q', r = 0$ o que verifica as condições do algoritmo da divisão.

Se $r' > 0$, temos,

$$a = -|a| = b(-q') - r' \iff a = b(-q') - b + b - r' \iff a = b(-q' - 1) + (b - r')$$

Como $0 \leq b - r' < b$, os inteiros $q = -q' - 1$ e $r = b - r'$, verificam as condições do algoritmo enunciado.

Considere $b < 0$, qualquer que seja a , pela parte anterior podemos determinar q' e r' , tais que,

$$a = |b|q' + r' \text{ e } 0 \leq r' < |b|$$

Quando $b < 0$, temos que $|b| = -b$, assim,

$$a = |b|q' + r' \iff a = (-b)q' + r' \iff a = b(-q') + r'$$

Portanto para os inteiros $q = -q'$ e $r = r'$, são válidas as condições do algoritmo. Agora provaremos a unicidade de q e r . Considere dois inteiros a e b , e que seja válida a equação $a = bq + r$ e $0 \leq r < |b|$, suponha que existem q' e $r' \in \mathbb{Z}$ tal que, $a = bq' + r'$ e $0 \leq r' < |b|$. Daí segue que $(a - bq) = r$ e que $(a - bq') = r'$. Queremos mostrar que $q = q'$ e que $r = r'$. Como r e r' são inteiros, podemos supor sem perda de generalidade que $r' \leq r$. Subtraindo as expressões temos:

$$r - r' = (a - bq) - (a - bq') \iff r - r' = b(q - q')$$

Como tanto r quanto r' são números menores que $|b|$, e $r' \leq r$, obtemos que $0 \leq r - r' < |b|$. E comparando $r - r' = b(q' - q)$, concluímos que $0 \leq b(q' - q) < |b|$. Assim $0 \leq (q - q') < 1$. Como q e q' são inteiros, esta desigualdade só é verificada quando $q' - q = 0$, ou seja $q = q'$.

Substituindo q e q' na equação $r - r' = b(q' - q)$ obtemos $r - r' = 0$, logo $r = r'$.

Portanto temos que existem únicos inteiros q e r tais que $a = bq + r$, com $0 \leq r < |b|$.

Lema 1 *Sejam a e b inteiros, $b \neq 0$, e sejam q e r , respectivamente, o quociente e o resto da divisão de a por b . Então $D(a, b) = D(b, r)$ em particular $\text{mdc}(a, b) = \text{mdc}(b, r)$.*

Demonstração

Dados $a, b \in \mathbb{Z}$, existem $q, r \in \mathbb{Z}$ tal que $a = bq + r$. Considere $x \in D(a, b)$. Então $x|a$ e $x|b$. Como $r = a - bq$ e x divide cada um dos somados, então $x|r$. Mostramos assim que $D(a, b) \subset D(b, r)$. A inclusão contrária segue de forma análoga. Donde resulta a igualdade dos conjuntos.

Se os conjuntos são iguais, seus máximos também são, isto é, $\text{mdc}(a, b) = \text{mdc}(b, r)$.

Algoritmo 2 (Algoritmo Euclidiano)

Considere a e b inteiros positivos com $a \geq b$. Para calcular o $\text{mdc}(a,b)$, o *Algoritmo Euclidiano* consiste em dividir a por b , achando o resto r_1 . Se $r_1 \neq 0$, dividimos b por r_1 , obtendo o resto r_2 . Se $r_2 \neq 0$, dividimos r_1 por r_2 , obtendo o resto r_3 , e assim por diante. Chamamos de *máximo divisor comum* o último resto diferente de zero desta sequência de divisões. Tal processo decorre do algoritmo da divisão e possui um fim, pois o resto é sempre menor que o divisor e em determinado momento tem que zerar. A tabela a seguir apresenta como obtermos tal algoritmo.

Tabela 1: Cálculo do Algoritmo Euclidiano

	q_1	q_2	q_3	\dots	q_{n-1}	q_n	q_{n+1}
a	b	r_1	r_2	\dots	r_{n-2}	r_{n-1}	$r_n = \text{mdc}(a, b)$
r_1	r_2	r_3	r_4	\dots	r_n	0	

Fonte: HEFEZ, 2013, P. 91

Vejamos então a mérito de exemplo o cálculo do $\text{mdc}(372, 161)$.

Tabela 2: Cálculo do $\text{mdc}(372, 161)$

	2	3	4	1	1	5
372	161	50	11	6	5	1
50	11	6	5	1	0	

Elaborado pelo autor

Assim temos que o $\text{mdc}(372,161)$ é igual a 1. Ou seja 372 e 161 são primos entre si.

1.3 Propriedades de Congruência

Podemos afirmar que a congruência módulo n é uma relação de equivalência, pois são satisfeitas as propriedades de reflexão, simetria e transitividade.

Propriedade 1 *Reflexiva*

$$a \equiv a(\text{mod } n).$$

Demonstração

Cosidere a e $n \in \mathbb{Z}$, temos que $a - a = 0$, como $n|0$ qualquer que seja n , então $n|a - a$. Portanto $a \equiv a(\text{mod } n)$.

Propriedade 2 *Simétrica*

Se $a \equiv b \pmod{n}$, então $b \equiv a \pmod{n}$.

Demonstração

Considere $a \equiv b \pmod{n}$ então $n|a - b$, existe $q \in \mathbb{Z}$ tal que $a - b = nq$, logo $b - a = -(a - b) = n(-q)$, Daí concluímos que $n|b - a$. Portanto $b \equiv a \pmod{n}$.

Propriedade 3 *Transitiva*

Se $a \equiv b \pmod{n}$ e $b \equiv c \pmod{n}$, então $a \equiv c \pmod{n}$.

Demonstração

Considere $a \equiv b \pmod{n}$ e $b \equiv c \pmod{n}$, então existem $k, m \in \mathbb{Z}$, tais que $a - b = kn$ e $b - c = mn$. Logo,

$$a - c = (a - b) + (b - c) = (k + m)n. \text{ Portanto } a \equiv c \pmod{n}.$$

Também temos que as propriedades a seguir são validas.

Propriedade 4 Se $a \equiv b \pmod{n}$ e $c \equiv d \pmod{n}$, então $a + c \equiv b + d \pmod{n}$.**Demonstração**

Considere $a, b, c, d \in \mathbb{Z}$, tais que $a \equiv b \pmod{n}$ e $c \equiv d \pmod{n}$. Como $n|a - b$ e $n|c - d$, existem k e $m \in \mathbb{Z}$. tais que $(a - b) = km$ e $(c - d) = mn$, assim $(a + c) - (b + d) = (k + m)n$. Portanto $a + c \equiv b + d \pmod{n}$.

Propriedade 5 Se $a \equiv b \pmod{n}$ e $c \equiv d \pmod{n}$, então $ac \equiv bd \pmod{n}$.**Demonstração**

Considere $a, b, c, d \in \mathbb{Z}$, tais que $a \equiv b \pmod{n}$ e $c \equiv d \pmod{n}$. Como $n|a - b$ e $n|c - d$, existem k e $m \in \mathbb{Z}$, tal que $a = b + kn$ e $c = d + mn$. Assim temos que $ac = (b + kn)(d + mn) = bd + n(bm + dk + nkm)$. Portanto $ac \equiv bd \pmod{n}$.

Propriedade 6 Se $\text{mdc}(n, c) = 1$, então $ac \equiv bd \pmod{n}$ implica $a \equiv b \pmod{n}$.**Demonstração**

Para demonstrar esta propriedade vamos primeiramente demonstrar o *Teorema de Euclides*.

Teorema 1 (Teorema de Euclides) Sejam a, b, c inteiros tais que $a|bc$. Se $\text{mdc}(a, b) = 1$, então $a|c$.

A proposição a seguir será utilizada para demonstração do *Teorema de Euclides*.

Proposição 1 *Sejam a, b inteiros, $d = \text{mdc}(a, b)$ e c um inteiro não nulo então:*

(i) $\text{mdc}(ac, bc) = d|c|$; e

(ii) Se $c|a$ e $c|d$, então $\text{mdc}\left(\frac{a}{c}, \frac{b}{c}\right) = \frac{d}{|c|}$.

Demonstração

Para (i), considere $d = \text{mdc}(a, b)$, temos então que $d|a$ e $d|b$, então existe $c \in \mathbb{Z}$, tal que $d|c|ac$ e $d|c|bc$, logo $\text{mdc}(ac, bc) = d|c|$. Agora, seja d' um inteiro tal que $d'|ac$ e $d'|bc$, da relação anterior temos que $d'|d|c|$.

Para (ii), Seja $d' = \text{mdc}\left(\frac{a}{c}, \frac{b}{c}\right)$, de (i) temos que ,

$$\text{mdc}(a, b) = \text{mdc}\left(\frac{a}{c} \cdot c, \frac{b}{c} \cdot c\right) = \text{mdc}\left(\frac{a}{c}, \frac{b}{c}\right) |c|, \text{ ou seja } d' = \frac{d}{|c|}.$$

Voltando para demonstração do *Teorema de Euclides*. Considere $\text{mdc}(a, b) = 1$, dai segue que $\text{mdc}(ac, bc) = 1|c|$. Então temos que $a|bc$, por hipótese. Consequentemente usando (ii) da proposição acima temos que $a|c|$, logo $a|c$.

Retomando a demonstração da propriedade enunciada temos que $\text{mdc}(n, c) = 1$, e que $ac \equiv bd \pmod{n}$, então $n|(ac - bc) \Rightarrow n|(a - b)c$. Pelo *Teorema de Euclides* $n|a - b$. Portanto $a \equiv b \pmod{n}$.

Propriedade 7 *Se $a \equiv b \pmod{n}$, então $a^m \equiv b^m \pmod{n}$, para todo inteiro positivo m .*

Demonstração

Vamos provar por indução sobre m . Temos que para $m = 1$ $a \equiv b \pmod{n}$.

Consederemos verdadeiro para $m = k$ com $k \in \mathbb{Z}$, temos então que $a^k \equiv b^k \pmod{n}$. Vamos mostrar que a propriedade vale quando $m = k + 1$, assim por hipótese de indução $a^k \equiv b^k \pmod{n}$ como $a \equiv b \pmod{n}$ usando a **Propriedade 5** temos que $aa^k \equiv bb^k \pmod{n} \iff a^{k+1} \equiv b^{k+1} \pmod{n}$. Portanto $a^m \equiv b^m \pmod{n}$, para todo $m \in \mathbb{Z}$.

Propriedade 8 *Se $a + c \equiv b + c \pmod{n}$, então $a \equiv b \pmod{n}$.*

Demonstração

Considere $a, b, c \in \mathbb{Z}$, tais que $a + c \equiv b + c \pmod{n}$, então existe $q \in \mathbb{Z}$ tal que $(a + c) - (b + c) = nq$, assim temos que $a + c - b - c = nq$, logo $a - b = nq$, Portanto $a \equiv b \pmod{n}$.

Propriedade 9 *Seja p um número primo e $a, b \in \mathbb{Z}$ tai que $p|ab$, então $p|a$ ou $p|b$.*

Demonstração

Considere que $p \nmid a$ então $\text{mdc}(p, a) = 1$, como $p|ab$ temos que existe um $k \in \mathbb{Z}$ tal que $p = kab \iff p = b(ka)$ com $ka = t \in \mathbb{Z}$, portanto $p|b$. Tal demonstração é análoga para $p \nmid b$.

Propriedade 10 *Sejam p um número primo e a_1, a_2, \dots, a_k números inteiros tais que $p|a_1 \cdot a_2 \cdot \dots \cdot a_k$. Então existe $n \in \mathbb{N}$, $1 \leq n \leq k$, tal que $p|a_n$.*

Demonstração

Se $p|a_1 \cdot a_2 \cdot \dots \cdot a_k$, pela **Propriedade 9**, $p|a_1$ ou $p|a_2 \cdot \dots \cdot a_k$. Se $p|a_1$, está provada a propriedade. Se $p|a_2 \cdot \dots \cdot a_k$, aplicando novamente a **Propriedade 9** temos que $p|a_2$ ou $p|a_3 \cdot \dots \cdot a_k$. Como temos um produto de k inteiros, o processo será repetido uma quantidade finita de vezes até encontrarmos o a_n , $n \in \mathbb{N}$, $1 \leq n \leq k$, que é divisível por p , provando assim a propriedade.

1.4 Teoremas

Os teoremas a seguir são necessários para demonstrar a funcionalidade do método RSA.

Lema 2 *Seja \mathbf{S} um ideal de \mathbb{Z} . Então, $\mathbf{S} = \{0\}$ ou existe um inteiro positivo m tal que $\mathbf{S} = m\mathbb{Z}$.*

Demonstração

Seja \mathbf{S} um ideal de \mathbb{Z} . Se $\mathbf{S} \neq \{0\}$, existe pelo menos um inteiro $a \neq 0$. Como \mathbf{S} um ideal de \mathbb{Z} então $-a \in \mathbf{S}$. Assim podemos afirmar que \mathbf{S} possui elementos positivos.

Considere agora \mathbf{S}^+ é igual a $\{\alpha \in \mathbf{S} \mid \alpha > 0\}$ é não-vazio e formado por elementos não negativos. Pelo Princípio da Boa Ordem, existe $m = \min \mathbf{S}$.

Como $m \in \mathbf{S}$, e \mathbf{S} é um ideal de \mathbb{Z} , então $mx \in \mathbf{S}$ para todo $x \in \mathbb{Z}$, isto garante que $m\mathbb{Z} \subseteq \mathbf{S}$.

Considere agora $\alpha \in \mathbf{S}$ pelo algoritmo da divisão existem $q, r \in \mathbb{Z}$ tais que $\alpha = mq + r$ em que $0 \leq r < m$.

Se $r \neq 0$, escrevendo $r = \alpha - mq$ como tanto α quanto m pertencem a \mathbf{S} e \mathbf{S} é um ideal de \mathbb{Z} , então $\alpha - m = r \in \mathbf{S}$. Como $0 \leq r < m$ isto contradiz a a minimalidade de r em \mathbf{S}^+ . Logo temos que $r = 0 \Rightarrow \alpha = mq \Rightarrow \alpha \in m\mathbb{Z}$.

Assim $\mathbf{S} \subseteq m\mathbb{Z}$ e como já mostramos a inclusão contrária, vale a igualdade.

Teorema 2 (Teorema de Bézout) *Sejam a, b inteiros e $d = \text{mdc}(a, b)$. Então, existem inteiros r e s tais que $d = ra + sb$.*

Demonstração

Seja $\mathcal{S} = \{ax + by | x, y \in \mathbb{Z}\}$.

Se α e $\beta \in \mathcal{S}$, então existem x_1, y_1 e $x_2, y_2 \in \mathbb{Z}$, tais que $\alpha = ax_1 + by_1$ e $\beta = ax_2 + by_2$, logo

$$\alpha + \beta = ax_1 + by_1 + ax_2 + by_2 = a(x_1 + x_2) + b(y_1 + y_2)$$

Seja agora $\alpha \in \mathcal{S}$ e dado um $c \in \mathbb{Z}$, então existem x e $y \in \mathbb{Z}$, tais que $\alpha = ax + by$, neste caso $c\alpha = c(ax + by) = a(cx) + b(cy)$.

Provamos assim que se $\alpha \in \mathcal{S}$ e $c \in \mathbb{Z}$ e $c\alpha \in \mathcal{S}$ então \mathcal{S} é um ideal de \mathbb{Z} .

Assim sendo, pelo **Lema 2** existe $m \in \mathbb{Z}$ tal que $\mathcal{S} = m\mathbb{Z}$.

Vamos mostrar que $m = \text{mdc}(a, b)$.

Note que $a = a \cdot 1 + b \cdot 0 \Rightarrow a \in \mathcal{S} \Rightarrow a = tm$, para algum $t \in \mathbb{Z}$. De modo análogo temos que $b \in \mathcal{S}$, então existe um $v \in \mathbb{Z}$ tal que $b = vm$.

Vimos então que $m|a$ e $m|b$, logo $m \in \mathbf{D}(a, b)$. Além disso $m \in \mathcal{S}$, assim existem $r, s \in \mathbb{Z}$ tal que $m = ar + bs$.

Seja agora $d' \in \mathbf{D}(a, b)$. então $d'|a$ e $d'|b$. Pela **Proposição 1**, temos que $d'|ar + bs = m$. Provamos então que $m \in \mathbf{D}(a, b)$ e que para todo $d' \in \mathbf{D}(a, b)$ temos que $d'|m$.

Portanto $m = \text{mdc}(a, b) = d'$ e $d' = m = ar + bs$ para $r, s \in \mathbb{Z}$.

Lema 3 *Dado $p \in \mathbb{Z}$, p primo, então p não divide $(p-1)!$ e nesse caso o $\text{mdc}(p, (p-1)!) = 1$.*

Demonstração

Considere que $p|(p-1)!$. Então como $(p-1)! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot (p-1)$ temos que $p|1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot (p-1)$.

Como p é primo, pela **Propriedade 10** existe $j \in \{1, 2, 3, 4, \dots, (p-1)\}$ tal que $p|j$, mas $j < p$, uma contradição.

Portanto p não divide $(p-1)!$ assim sendo o $\text{mdc}(p, (p-1)!) = 1$.

Teorema 3 (Teorema de Fermat) *Sejam p um primo e a um inteiro tal que p não divide a . Então $a^{p-1} \equiv 1 \pmod{p}$.*

Demonstração

Consideremos \mathcal{S} o conjunto dos primeiros $p-1$ múltiplos de a .

$$\mathcal{S} = \{a, 2a, 3a, 4a, \dots, (p-1)a\}$$

Dados $xa, ya \in \mathcal{S}$, se ocorresse de $xa \equiv ya \pmod{p}$, como por hipótese $\text{mdc}(a, p) = 1$ teríamos então pela **Propriedade 6** temos que $x \equiv y \pmod{p}$ o que não ocorre pois $x, y \in \mathcal{B} = \{1, 2, 3, 4, 5, \dots, (p-1)\}$ que é um sistema completo de resíduos módulo p , então $xa \not\equiv ya \pmod{p}$, para todo $xa, ya \in \mathcal{S}$

Temos ainda que nenhum elemento de \mathcal{S} será congruente a zero módulo p , pois se $xa \in \mathcal{S}$ é tal que $xa \equiv 0 \pmod{p}$ teríamos que $p|xa$ e como $\text{mdc}(p, a) = 1$ teríamos que $p|x$, absurdo, pois $1 \leq x \leq p-1$.

Como \mathcal{B} é um sistema completo de resíduos módulo p , então cada elemento de \mathcal{S} é congruente módulo n a um elemento de \mathcal{B} , em uma ordem conveniente. Temos então $p-1$ congruências na forma:

$$\begin{aligned} a &\equiv x_1 \pmod{p} \\ 2a &\equiv x_2 \pmod{p} \\ &\cdot \\ &\cdot \\ &\cdot \\ (p-1)a &\equiv x_{p-1} \pmod{p}. \end{aligned}$$

Como $x_1, x_2, \dots, x_{p-1} \in \mathcal{B}$.

Usando a **Propriedade 5**, temos

$$a \cdot 2a \cdot \dots \cdot (p-1)a \equiv 1 \cdot 2 \cdot \dots \cdot (p-1) \pmod{p}.$$

Assim temos que, $a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}$. Mostramos no **Lema 3** que $\text{mdc}(p, (p-1)!) = 1$. Portanto pela **Propriedade 6** concluímos que $a^{p-1} \equiv 1 \pmod{p}$.

Lema 4 *Sejam p, q e r inteiros positivos e suponhamos que p e q são primos entre si.*

(i) Se $q|pr$ então $q|r$. (ii) Se $p|r$ e $q|r$ então $pq|c$

Demonstração

Provaremos inicialmente que (i) é válida, temos também por hipótese que p e q são primos entre si, ou seja $\text{mdc}(p, q) = 1$. O *Teorema de Bézout* nos garante que existem $\alpha, \beta \in \mathbb{Z}$ tais que $\alpha a + \beta b = 1$. Consideremos $r \in \mathbb{Z}$ tal que $\alpha ac + \beta bc = c$, é evidente que a segunda parcela da soma é divisível por q , como de (i) temos que $q|pr$, logo o a primeira parcela também é divisível por q . Portanto $q|r$.

Podemos provar (ii) a partir de (i). De fato, se $p|r$, então existe $t \in \mathbb{Z}$ tal que $r = pt$. Mas $q|r$, como p e q são primos entre si, segue da proposição (i) que $t|q$. Assim teremos que existe um $k \in \mathbb{Z}$ tal que $t = bk$. Portanto $r = pt = p(qk) = (pq)k \Rightarrow pq|c$.

Lema 5 *Se p é primo, então $\phi(p) = (p - 1)$.*

Demonstração

Temos que p é primo, assim $\mathcal{A} = \{1, 2, 3, 4, 5, \dots, (p - 1)\}$ forma um sistema de resíduos completo módulo p , assim temos que p possui $p - 1$ números interiores relativamente primos a ele. Portanto $\phi(p) = (p - 1)$.

Lema 6 *Se p e q primos, então $\phi(pq) = (p - 1)(q - 1)$.*

Demonstração

Temos que p, q são primo, assim $\mathcal{A} = \{1, 2, 3, 4, 5, \dots, (p - 1)\}$ forma um sistema de resíduos completo módulo p e $\mathcal{B} = \{1, 2, 3, 4, 5, \dots, (q - 1)\}$. Do **Lema 5** temos que $\phi(p) = (p - 1)$ e $\phi(q) = (q - 1)$. Considere então $n = pq$, como $\text{mdc}(p, q) = 1$ existem $p - 1$ números que dividem n e $q - 1$ números que dividem n . Logo,

$$\begin{aligned}\phi(n) &= (n - 1) - (p - 1) - (q - 1) = n - p - q + 1 = pq - p - q + 1 \\ &\iff \phi(pq) = (p - 1)(q - 1).\end{aligned}$$

2 Criptografia

A criptografia é o estudo de métodos para a codificação e decodificação de mensagens, o termo originou-se do grego *cryptos* que significa secreto, oculto.

Segundo Singh (2004), o primeiro documento que usou uma cifra de substituição aparece em Roma nas guerras de Gália de Júlio César, chamadas de cifras de César, que consistia em substituir as letras do alfabeto romano por letras gregas e assim a mensagem ficava incompreensível para o inimigo.

Ainda segundo Singh (2004) na antiguidade, os estudiosos árabes inventaram a Criptoálise, ciência que permite decifrar uma mensagem sem conhecer a chave. Eles utilizavam um alfabeto cifrado, que era um simples rearranjo de alfabeto, conhecido também como cifra de substituição monoalfabética, que consiste em substituir cada letra por um símbolo. Este método tornou-se vulnerável, pois conhecendo no idioma o qual a mensagem foi escrita, eram utilizados métodos estatísticos utilizando as letras mais frequentes no idioma e comparando com o símbolo que mais aparecia na mensagem codificada de forma sucessiva até a mensagem ser decifrada.

Segundo Hefez (2013), a evolução da criptografia foi no sentido de não mais ocultar fisicamente as mensagens, mas usar estratégias para ocultar o seu significado das pessoas que não fossem as legítimas destinatárias das mesmas, de modo que pudessem ser veiculadas através de um canal público de comunicação.

A principal fraqueza dos sistemas criptográficos por substituição simples, como as cifras de César, é que em um texto de uma determinada língua as letras do alfabeto ocorrem com frequências distintas, além de existirem regras rígidas de contato entre letras, como por exemplo, na língua portuguesa, a letra *q* vem sempre seguida pela letra *u*.

Segundo Hefez (2013), para evitar a quebra de um código por análise de frequência, há uma outra vertente de sistemas criptográficos que se baseia na transposição, formando anagramas da mensagem original. Por exemplo, uma mensagem com 100 letras dá origem a $100!$ permutações distintas das letras, um número que torna praticamente impossível de ser decifrada a mensagem se não possuímos a chave para tal. A combinação do método de substituição de letras e transposição, deu origem a sistemas criptográficos mais sofisticados.

Diz Singh (2004) que na segunda guerra mundial, a forma de ciframento que se tornou popular foi o microponto. agentes alemães operando na América Latina, reduziam fotograficamente uma página de texto até transformá-la num ponto com menos de um milímetro de diâmetro. O microponto era ocultado sobre o ponto final de uma carta aparentemente inofensiva e para saber o conteúdo era necessária uma lupa. Nesta mesma época os britânicos construíram o primeiro computador programável que decifrava a cifra alemã Lorenz usada para estabelecer a comunicação entre Hitler e seus generais. O que veio a ser o início da criptografia moderna.

2.1 Método de Criptografia RSA

Um artigo publicado em 1976 por Whitfield Diffie e Martin Hellman apontava como necessidade para os avanços tecnológicos a criptografia das informações antes de serem enviadas. O grande problema era que a chave não poderia ser enviada por e-mail ou via correio, pois poderia ser interceptada. Os dois cientistas da computação sugeriram então um novo método para que a chave fosse enviada de forma segura. Tal forma consistia em que todas as informações necessárias para a troca fossem disponibilizadas publicamente. Pensou-se, assim, em um criptossistema de chave pública.

Segundo Okumura (2014), um criptossistema de chave pública deve conter um esquema público de codificação E e uma esquema privado de decodificação D , de modo que E e D sejam obtidos de modo fácil, e para uma mensagem M temos, $D(E(M)) = M = E(D(M))$.

Pouco depois da publicação do artigo de 1976, três estudantes do MIT (Massachusetts Institute of Technology) passaram a buscar um novo tipo de criptografia que satisfizesse o criptossistema proposto por Diffie e Hellman. Assim eles estabeleceram a seguinte dinâmica: dois deles, Ronald e Adi, trabalhavam em ideias de como "esconder" uma mensagem e Leonard trabalhava em adivinhar a técnica utilizada. Depois de diversas tentativas, certo dia Ronald trouxe um algoritmo que Leonard não conseguiu quebrar. Assim surgiu o RSA em homenagem a seus criadores Ronald Rivest, Adi Shamir e Leonard Adleman, método que permanece seguro desde então.

O método RSA foi desenvolvido em 1978 e nos últimos 40 anos pesquisadores encontraram algumas fraquezas na implementação do algoritmo, mas que foram sendo corrigidas. O único método de criptografia de chave pública que resistiu a mais de 30 anos de ataques. A criptografia RSA é considerada a melhor alternativa para criptografar transações com cartões de crédito via internet, autenticação de chamadas telefônicas e segurança de e-mails.

Segundo Coutinho (2009), para implementar o método de criptografia RSA, precisamos escolher dois números primos p e q muito grandes. Para codificar uma mensagem, usamos $n = pq$, e para decodificar precisamos conhecer p e q . A segurança do método vem do fato de que é difícil fatorar n para descobrir p e q , já que são números muito grandes.

3 Implantação da Criptografia RSA

Para a implantação do método de criptografia RSA é necessário o desenvolvimento da pré-codificação, codificação e decodificação deste método.

3.1 Pré Codificação

Em um primeiro momento para usarmos a criptografia RSA, devemos converter uma mensagem em uma sequência de números.

Considere a seguinte tabela de conversão como exemplo.

Tabela 3: Conversão de letras em números

A	B	C	D	E	F	G	H	I	J	K	L	M
10	11	12	13	14	15	16	17	18	19	20	21	22
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
23	24	25	26	27	28	29	30	31	32	33	34	35

Fonte: COUTINHO, 2009, p.181

O espaço entre duas palavras será substituído pelo número 36.

Segundo Coutinho (2009), a vantagem de fazer cada letra corresponder a um número de dois algarismos é que isso evita ambiguidade. O que aconteceria se fizéssemos A corresponder ao número 1, B ao 2 e assim por diante, neste caso 12 poderia ser AB ou L, que é a décima segunda letra do alfabeto.

Em sequência com a pré codificação, vamos determinar dois primos p e q , que serão denominados de parâmetros RSA e chamamos de módulo RSA $n = pq$. A última fase de pré-codificação, consiste em separarmos a sequência numérica obtida por meio da conversão de letras em números. Usaremos como exemplo a frase "*Eu amo Matemática*".

Na frase citada como exemplo, ao convertermos as letras em números obtemos a sequência numérica:

1430991022249922102914221029191210

Após determinamos a sequência numérica devemos quebrá-la em blocos. Podemos quebrar a sequência numérica nos seguintes blocos:

14-309-9102-224-992-2102-91-422-102-91-91-210

Coutinho (2009) afirma que a maneira de se obter os blocos não é única e não precisa ser homogênea (todos os blocos com o mesmo número de dígitos), mas certos cuidados devem ser tomados como por exemplo, é necessário que o bloco não comece com o zero porque isso traria problemas na hora de decodificar.

Também segundo Coutinho (2009), podemos observar que os blocos em que a mensagem foi quebrada não correspondem a nenhuma unidade linguística (palavra, letra ou frase). O que torna a decodificação por contagem de frequência essencialmente impossível.

3.2 Codificação e Decodificação

Para iniciarmos o processo de codificação é necessário utilizarmos n , que é o produto dos primos p e q , e precisamos de um inteiro positivo h que seja inversível módulo $\phi(n)$. ou seja $\text{mdc}(h, \phi(n)) = 1$. Obtemos o $\phi(n)$, conhecendo p e q por meio da expressão:

$$\phi(n) = (p - 1)(q - 1).$$

O par (n, h) é chamado de *chave de codificação* do sistema RSA.

Codificaremos cada bloco separadamente, e a mensagem codificada será a sequência dos blocos codificados. Considere então b , um bloco da mensagem codificada, Temos que $C(b)$ é o resto da divisão de b^h por n , isto é:

$$C(b) \equiv b^h \pmod{n}.$$

Para decodificar cada bloco b , precisamos de n e do inverso de h módulo $\phi(n)$, que denotaremos por d , ou seja:

$$hd \equiv 1 \pmod{\phi(n)}.$$

O par (n, d) é chamado de *chave de decodificação* do sistema RSA.

Considere $a = C(b)$ um bloco da mensagem codificada, assim $D(a)$ será o resultado da decodificação. Temos que $D(a)$ é o resto da divisão de a^d por n , ou seja:

$$D(a) \equiv a^d \pmod{n}.$$

Deste modo, para obtermos d , sendo conhecidos h e $\phi(n)$, basta aplicarmos o *Teorema de Bézout*, Sabemos que $\text{mdc}(hd, \phi(n)) = 1$, logo $1 = hd - k\phi(n)$ para $k \in \mathbb{Z}$. Portanto decodificando os blocos da mensagem codificada podemos encontrar a mensagem original, ou seja, $D(C(b)) = b$, assim para decodificarmos uma mensagem não é necessário conhecermos os primos p e q , basta conhecermos n e d .

Voltaremos ao exemplo da frase da seção anterior, "*Eu amo Matemática*".

A título de exemplo vamos quebrar a sequência numérica nos seguintes blocos:

14-30-36-102-224-36-22-10-29-142-210-29-191-210

Considerando p e q , 13 e 17 respectivamente, dois números primos. E n igual a 221. Assim,

$$\begin{aligned}\phi(n) &= (p-1)(q-1) \\ \phi(n) &= (13-1)(17-1), \text{ então} \\ \phi(n) &= 192\end{aligned}$$

Deste modo podemos obter h , inteiro positivo inversível módulo $\phi(n)$. Consideraremos $h = 5$.

Após determinarmos todos os valores necessários para o processo de codificação podemos então construir os $C(b)$ (blocos codificados) a serem enviados ao receptor, onde $C(b)$ é o resto da divisão de b^h por n . Conforme demonstra a tabela a seguir.

Tabela 4: Blocos Codificados

b	14	30	36	102	224	36	22
C(b)	131	166	134	85	22	134	133
b	10	29	142	210	29	191	210
C(b)	108	139	194	58	139	55	58

Elaborado pelo autor

Assim a sequência numérica a ser enviada ao receptor é:

131-166-134-85-22-134-133-108-139-194-58-139-55-58

O processo de decodificação deve ser feito pelo receptor. Assim para decodificar cada bloco $C(b)$ recebido, precisamos de $n = 221$ e do inverso de h módulo $\phi(n)$ que denotado por d . onde, $5d = 1 \pmod{192}$. Portanto podemos obter d pelo algoritmo da divisão, então dividindo $\phi(n)$ por 5, temos que $192 = 5 \cdot 38 + 2$, podemos simplificar a equação anterior e obtermos $96 = 5 \cdot 19 + 1$, logo, $1 = 96 - 5 \cdot 19$, logo o inverso de 5 módulo 192 é -19, como precisamos que d seja positivo temos que $d = 96 - 19 = 77$ que é o menor inteiro positivo congruente a -19 módulo 192.

Considerando cada bloco $a = C(b)$, podemos obter $D(a)$, (blocos decodificados) onde $D(a)$ é resto da divisão de a^d por n , conforme apresenta a tabela a seguir.

Tabela 5: Blocos Decodificados

a	131	166	134	85	22	134	133
D(a)	14	30	36	102	224	36	22
a	108	139	194	58	139	55	58
D(a)	10	29	142	210	29	191	210

Elaborado pelo autor

Portanto a sequência numérica obtida será:

Retomando assim a frase anteriormente codificada, "*Eu amo Matemática*".

3.3 Descrição matemática do método

Considere um sistema RSA de parâmetros p e q , com $n = pq$. então n e h serão as chaves de codificação, e n e d , as chaves de decodificação. Precisamos então verificar se b (bloco a ser codificado) é um inteiro e $1 \leq b \leq n - 1$, então $D(C(b)) = b$, ou seja $D(C(b)) \equiv b \pmod{n}$, ja que são inversíveis módulo $\phi(n)$.

Temos que $D(C(b)) \equiv (b^h)^d \equiv b^{hd} \pmod{n}$, sabendo que $n = pq$, podemos calcular de forma reduzida b^{hd} módulo p e módulo q , como o cálculo para ambos é análogo visto que p e q são primos, vamos calcular o caso de b^{hd} módulo p .

Como d é inverso de h módulo $\phi(n)$, pelo (*Teorema de Bézout*) temos que $hd = 1 + k\phi(n) = 1 + k(p - 1)(q - 1)$, para algum $k \in \mathbb{Z}$, Logo, $b^{hd} \equiv b(b^{p-1})^{k(q-1)} \pmod{p}$.

Suponhamos que p não divide b , então pelo *Teorema de Fermat*, $b^{p-1} \equiv 1 \pmod{p}$, ou seja $b^{hd} \equiv b \pmod{p}$

Como $b^{hd} \equiv b \pmod{p}$, de modo análogo temos que $b^{hd} \equiv b \pmod{q}$, segue então que $b^{hd} - b$ é divisível por p e q . Mas p e q são primos distintos, isto é $\text{mdc}(p, q) = 1$, pelo **Lema 4** temos que $pq | b^{hd} - b$. Como $n = pq$, concluímos que $b^{hd} \equiv b \pmod{n}$. Portanto $D(C(b)) \equiv b \pmod{n}$.

3.4 Segurança do sistema RSA

A criptografia RSA é um método de criptografia de chave pública. A chave de codificação, o par (n, h) , é dito a chave pública do sistema. Por isso, o RSA só será seguro se for difícil de encontrar d a partir de n e h .

Para obtermos d , utilizamos $\phi(n)$ e h , mas para obtermos $\phi(n)$, devemos necessariamente ter p e q , que é a fatoração de n . Deste modo para quebrar o código, temos que conseguir fatorar n , o que é um problema muito difícil considerando n um número grande de 10^{60} casas decimais, ou seja quanto maior p e q mais seguro o processo de criptografia RSA.

E caso não tenhamos d e desejamos obter b a partir de $C(b) \equiv b^h \pmod{n}$, é praticamente impossível se n for grande. Na verdade, acredita-se que quebrar o RSA e fatorar n são problemas equivalentes.

Segundo Freitas, Souza e Agustini (2004), devemos tomar alguns cuidados, pois se p e q forem pequenos, torna-se fácil encontrá-los. Ou se, mesmo grandes, $|p - q|$ for pequeno torna-se fácil achá-los a partir de n .

4 Programa de Criptografia RSA

Neste capítulo apresentamos um programa de criptografia RSA em linguagem *Java*, desenvolvido no sistema *NetBeans IDE 8.2*, para ilustração do processo de criptografia RSA descrito no capítulo anterior. Ressaltamos que esse programa foi elaborado apenas para o estudo do processo de criptografia RSA. Sendo assim programamos apenas o processo de codificação e decodificação de mensagens.

O processo de criptografia apresentado neste trabalho de pesquisa pode ser desenvolvido em diversas linguagens, pois os conceitos matemáticos por trás da criptografia RSA são aplicáveis independente da linguagem.

Adotamos a tabela ASCII, código padrão americano para o intercâmbio de informações, para executar a conversão de letras em números, tal tabela encontra-se disponível em: <http://www.theasciicode.com.ar/>

Tabela 6: Tabela ASCII

ASCII control characters			ASCII printable characters			Extended ASCII characters										
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	à	192	Ł	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ú	161	í	193	ł	225	ô
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	õ
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ü	195	ł	227	ö
04	EOT	(End of Trans.)	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ø
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	†	229	ó
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	á	166	ª	198	ā	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Ă	231	þ
08	BS	(Backspace)	40	(72	H	104	h	136	ê	168	¿	200	Ē	232	þ
09	HT	(Horizontal Tab)	41)	73	I	105	i	137	ë	169	®	201	ƒ	233	Ů
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	¬	202	±	234	Ű
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	í	171	½	203	ƒ	235	Ū
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¼	204	ƒ	236	ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ï	173	⅓	205	=	237	ÿ
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ā	174	«	206	ƒ	238	˘
15	SI	(Shift In)	47	/	79	O	111	o	143	Ă	175	»	207	ƒ	239	˙
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	ƒ	208	ð	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	ƒ	209	Đ	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	ƒ	210	É	242	…
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ó	179	ƒ	211	Ê	243	¼
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ô	180	ƒ	212	Ë	244	½
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	õ	181	Ā	213	Ī	245	¾
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	ù	182	Ă	214	Ī	246	+
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ú	183	Ā	215	Ī	247	.
24	CAN	(Cancel)	56	8	88	X	120	x	152	ý	184	©	216	Ī	248	°
25	EM	(End of medium)	57	9	89	Y	121	y	153	ÿ	185	ƒ	217	Ī	249	˘
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ů	186	ƒ	218	Ī	250	.
27	ESC	(Escape)	59	;	91	[123	{	155	Ű	187	ƒ	219	Ī	251	˘
28	FS	(File separator)	60	<	92	\	124		156	ƒ	188	ƒ	220	Ī	252	˘
29	GS	(Group separator)	61	=	93]	125	}	157	ø	189	€	221	Ī	253	˘
30	RS	(Record separator)	62	>	94	^	126	~	158	ˆ	190	Ÿ	222	Ī	254	˘
31	US	(Unit separator)	63	?	95	_			159	f	191	ƒ	223	Ī	255	nbsp
127	DEL	(Delete)														

Fonte: THE ASCII CODE (2018)

Podemos verificar na figura a seguir, ao executarmos o programa, a tela inicial que é apresentada ao usuário.

Figura 1: Tela Inicial do Programa RSA

The screenshot shows a window titled "RSA" with standard window controls. The main heading is "Criptografia RSA: O Programa". There are three main sections:

- Top Left (Parametros RSA):** Input fields for "Número p:" and "Número q:", and a button labeled "Gerar p e q aleatórios".
- Top Right (Parametros RSA):** Input fields for "Módulo n:", "t(n)=(p-1)(q-1):", "Clave Publica h:", and "Clave Privada d:", with a "Calcular" button.
- Bottom (Parametros RSA):** Radio buttons for "CODIFICAR" (selected) and "DECODIFICAR". A text area for input with the prompt "Digite ou cole o texto a ser CODIFICADO ou DECODIFICADO", an "EXECUTAR" button, a result area with the prompt "Resultado: Texto CODIFICADO ou DECODIFICADO", and an "Apagar tudo" button.

Elaborado pelo autor

No primeiro quadro de parametro RSA o usuário tem a possibilidade de digitar p e q , ou gerar aleatoriamente tais números. no segundo quadro ao clicar em calcular, o usuário obtém, as chaves de codificação e decodificação, $n = pq$, $\phi(n) = (p - 1)(q - 1)$, aqui respresentada por $t(n) = (p - 1)(q - 1)$, h , número inversível módulo n e d , inverso de h módulo $\phi(n)$.

Vejamos na figura a seguir o processo de codificação para frase "*Eu amo matemática*", a título de exemplo.

Figura 2: Processo de codificação



Elaborado pelo autor

Utilizamos no exemplo apresentado acima, $p = 524287$ e $q = 8191$, gerando $n = 4294434817$, $\phi(n) = 4293902340$, $h = 431219$ e $d = 4167928859$. Ao executarmos a codificação para a frase "Eu amo matematica". obtemos o bloco codificado:

1121258288 1345553791 532479 730035935 916930775 4062243863 532479 916930775
 730035935 990265767 3079195541 916930775 730035935 990265767 2443563781
 699540331 730035935

Para decodificar tal bloco precisamos então executar o processo inverso, conforme mostra figura a seguir.

Figura 3: Processo de decodificação



Elaborado pelo autor

Retomando assim a mensagem original *"Eu amo matematica"*.

4.1 Código Java do Processo de Criptografia RSA

Para aplicação adequada do método de criptografia RSA necessitamos determinar dois pares de chaves (n, h) par de codificação e (n, d) par de decodificação. O algoritmo a seguir demonstra tal aplicação utilizando-se da lógica matemática apresentada no capítulo anterior.

```
//módulo RSA.
    n = p.multiply(q);
//função  $\phi(n)$ .
    totient = p.subtract(BigInteger.valueOf(1));
    totient = totient.multiply(q.subtract(BigInteger.valueOf(1)));
//Obtendo h inversível módulo n.
    do h = new BigInteger(2 * tamPrimo, new Random());
        while((h.compareTo(totient) != -1) ||
```

```

        (h.gcd(totient).compareTo(BigInteger.valueOf(1)) != 0));
// Obtendo d inverso de h modulo  $\phi(n)$ .
        d = h.modInverse(totient);

```

O processo de codificação pode ser desenvolvido ao considerarmos b um bloco a ser codificado e $C(b)$, um bloco da mensagem codificada, Temos que $C(b)$ é o resto da divisão de b^h por n . o algoritmo a seguir desenvolve este processo.

```

        int i;
// Determinando os blocos para codificação, cada bloco será correspondente a um
caracter da mensagem.
        byte[] temp = new byte[1];
        byte[] digitos = mensagem.getBytes();
        BigInteger[] bigdigitos = new BigInteger[digitos.length];
        for(i=0; i<bigdigitos.length;i++){
            temp[0] = digitos[i];
            bigdigitos[i] = new BigInteger(temp);
        }
        BigInteger[] encriptado = new BigInteger[bigdigitos.length];
//Processo de codificação.
        for(i=0; i<bigdigitos.length; i++)
// Codificação do blocos.
            encriptado[i] = bigdigitos[i].modPow(h,n);
        return(encriptado);

```

E por fim para gerar o processo de decodificação onde cada bloco $a = C(b)$, podemos obter $D(a)$, (blocos decodificados) onde $D(a)$ é resto da divisão de a^d por n ,

```

        for(int i=0; i<decodificado.length; i++)
//Processo de decodificação.
            decodificado[i] = encriptado[i].modPow(d,n);
        char[] charArray = new char[decodificado.length];
//Retornando a mensagem original.
        for(int i=0; i<charArray.length; i++)
            charArray[i] = (char) (decodificado[i].intValue());
        return(new String(charArray));

```

Tal algoritmo desenvolve o processo matemático por trás do método RSA descrito anteriormente e pode ser verificado conforme exemplo anteriormente desenvolvido.

5 Considerações Finais

A busca por um sistema de codificação de mensagens durante o desenvolvimento da informatização levou pesquisadores, tais como os citados neste trabalho de conclusão de curso, a desenvolverem um sistema de criptografia seguro para seus usuários.

O método de criptografia RSA surgiu por causa desta necessidade e atualmente é o sistema de chaves públicas mais utilizado, visto que os conceitos matemáticos que envolvem o sistema de codificação o tornam o mais seguro dos sistemas de criptografia dos últimos 40 anos.

Podemos verificar no presente trabalho que há aplicações que utilizam o estudo de Teoria dos Números por trás do método RSA. Tal sistema é uma das mais curiosas aplicações de números primos e de suas propriedades, pois fugiu ao uso comum, mais abstrato, direcionando para algo mais concreto e prático.

No decorrer do trabalho foi verificado que são válidas certas propriedades matemáticas envolvendo números primos, que foram demonstradas e aplicadas para o desenvolvimento do método RSA. Foi criado também um programa que possibilitou confirmar a validade dos cálculos que envolvem esse método.

O trabalho em questão é apenas mais uma contribuição para os estudos da Matemática e sua aplicação na Criptografia, sendo assim está aberto a críticas e considerações que o auxiliem a expandir e melhorar a pesquisa.

Referências

COUTINHO, Severino Collier. **Números Inteiros e Criptografia RSA**. 2. ed. Rio de Janeiro: Impa, 2009. 226 p.

DEVMEDIA. **Calculadora Java: Criação de aplicações em Java utilizando Swing**. São Paulo, 2012. Disponível em: <<https://www.devmedia.com.br/calculadora-java-criacao-de-aplicacoes-em-java-utilizando-swing/26007>>. Acesso em: 12 jun. 2018.

FLOSE, Vania Batista Schunck. **Criptografia e Curvas Elípticas**. 2011. 55 f. Dissertação (Mestrado) - Curso de Mestrado Profissional em Matemática - Temática Universitária, Instituto de Geociências e Ciências Exatas., Universidade Estadual Paulista - Unesp, Rio Claro, 2011.

FREITAS, Hélen Cristina de; SOUSA, Angélica Silva de; AGUSTINI, Edson. Um Enfoque Computacional da Criptografia RSA. **Famat em Revista**, Uberlândia, v. 1, n. 3, p.121-136, set. 2004. Semestral. Disponível em: <http://www.portal.famat.ufu.br/sites/famat.ufu.br/files/Anexos/Bookpage/Famat_Revista_03.pdf>. Acesso em: 12 out. 2017.

GUJ, Programação Java. **Criptografia RSA em Java**. São Paulo, 2007. Disponível em: <<http://www.guj.com.br/t/criptografia-rsa-em-java/7354>>. Acesso em: 10 jun. 2018.

HEFEZ, Abramo. **Aritmética**. Rio de Janeiro: Sbm, 2013. 338 p.

MYDLARZ, Dariusz. **RSA em Java**. São Paulo: Github, 2018. Disponível em: <<https://gist.github.com/dmydlarz/32c58f537bb7e0ab9ebf>>. Acesso em: 12 jun. 2018.

MILIES, Francisco César Polcino; COELHO, Sônia Pitta. **Números: Uma Introdução à Matemática**. 2. ed. São Paulo: Usp, 2006. 248 p.

MORIMOTO, Ricardo Minoru. **Números Primos: Propriedades, Aplicações e Avanços**. 2014. 63 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação - Mestrado Profissional em Matemática, Instituto de Geociência e Ciências Exatas, Universidade Estadual Paulista - Unesp, Rio Claro, 2014.

OKUMURA, Mirella Kiyu. **Números primos e criptografia RSA**. 2014. 41 f. Dissertação (Mestrado) - Curso de Programa de Mestrado Profissional em Matemática, Instituto de Ciências Matemáticas e de Computação - Icmc, Usp, São Carlos, 2014.

PRINCETON. **RSA.Java** São Paulo, 2018. Disponível em: <<https://introcs.cs.princeton.edu/java/99crypto/RSA.java>>. Acesso em: 12 jun. 2018.

RIVEST, R. L., SHAMIR, A. e ADLEMAN, L. M. **A method for obtaining digital signatures and public-key cryptosystems**. Commun. ACM 21, 2 (1978), p.120–126.

SINGH, Simon. **O livro dos Códigos**. São Paulo: Record, 2004. 446 p.

THE ASCII CODE. **ASCII**, 2018. Disponível em: <<http://www.theasciicode.com.ar/>>