UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE MATEMÁTICA

RAPHAEL VINÍCIUS GONÇALVES CAMARGO

CRIPTOGRAFIA RSA: ASPECTOS HISTÓRICOS E MATEMÁTICOS

SÃO CARLOS
2021

RAPHAEL VINÍCIUS GONÇALVES CAMARGO

CRIPTOGRAFIA RSA: ASPECTOS HISTÓRICOS E MATEMÁTICOS

Monografia apresentada ao Curso de Licenciatura em Matemática da Universidade Federal de São Carlos.

Orientador: Prof. Dr. João Carlos Vieira Sampaio

SÃO CARLOS
2021

Camargo, Raphael Vinícius Gonçalves

Criptografia RSA: aspectos históricos e matemáticos / Raphael Vinícius Gonçalves Camargo -- 2021.
44f.

TCC (Graduação) - Universidade Federal de São Carlos, campus São Carlos, São Carlos
Orientador (a): João Carlos Vieira Sampaio
Banca Examinadora: João Carlos Vieira Sampaio, Waldeck Schützer, Wladimir Seixas
Bibliografia

1. Criptografia RSA. 2. Números primos. 3. Congruência módulo m. I. Camargo, Raphael Vinícius Gonçalves. II. Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática (SIn)

DADOS FORNECIDOS PELO AUTOR

Bibliotecário responsável: Ronildo Santos Prado - CRB/8 7325

FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS

COORDENAÇÃO DOS CURSOS DE GRADUAÇÃO EM MATEMÁTICA - CCM/CCET
Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905
Telefone: (16) 33518221 - http://www.ufscar.br

DP-TCC-FA nº 17/2021/CCM/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

RAPHAEL VINÍCIUS GONÇALVES CAMARGO

CRIPTOGRAFIA RSA: ASPECTOS HISTÓRICOS E MATEMÁTICOS

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 24 de junho de 2021

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	João Carlos Vieira Sampaio
Membro da Banca 1	Waldeck Schützer
Membro da Banca 2	Wladimir Seixas

Documento assinado eletronicamente por **Wladimir Seixas**, **Professor do Magistério Superior**, em 11/08/2021, às 12:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Waldeck Schutzer**, **Professor do Magistério Superior**, em 13/08/2021, às 19:10, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Joao Carlos Vieira Sampaio**, **Professor do Magistério Superior**, em 15/08/2021, às 18:00, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufscar.br/autenticacao, informando o código verificador **0463613** e o código CRC **5811F80B**.

Referência: Caso responda a este documento, indicar expressamente o Processo nº 23112.011088/2021-66

SEI nº 0463613

Modelo de Documento: Grad: Defesa TCC: Folha Aprovação, versão de 02/Agosto/2019

AGRADECIMENTOS

Agradeço a Deus pela vida, porque tem sido meu refúgio na angústia, porque tem sido bom desde os tempos que não sou capaz de recordar e por ter me sustentado até aqui.

Agradeço a meus pais pelo esforço sem medidas que despenderam voluntariamente para que eu pudesse seguir um caminho muito mais fácil na minha formação, por muitas vezes abdicarem de si mesmos em meu favor, especialmente minha falecida mãe.

Aos meus irmãos que fizeram a minha vida brilhar cada vez mais e que se tornaram tão necessários e presentes quanto poderiam ser.

Aos meus amigos e amigas que se tornaram irmão e irmãs, sem os quais a caminhada teria sido extremamente difícil, pelos momentos que me consolaram, pelos risos e lágrimas que compartilhamos.

Enfim, agradeço às pessoas que acreditaram em mim, me apoiaram e encorajaram, ainda quando eu mesmo não conseguia fazê-lo.

RESUMO

Este trabalho estuda aspectos da criptografia de chave pública RSA e sua relação com a teoria dos números, em especial com as propriedades dos números primos. A metodologia apoia-se na pesquisa bibliográfica. O estudo é baseado na análise dos textos bibliográficos. Da análise dos textos ficam evidentes os pontos fortes da criptografia de chave pública RSA, também conhecida como criptografia assimétrica RSA. Como pontos fortes, pode-se citar a maior segurança sobre ataques e quebra do código por usar chaves grandes, mesmo usando computadores supermodernos e com processamento de dados ultraveloz e gerenciamento de chaves para todos os usuários do sistema. A principal desvantagem é a lentidão no processo de criptografia, por usar algoritmos geralmente mais complexos e chaves grandes.

Palavras-chave: Criptografia RSA. Números primos. Congruência módulo *m*.

ABSTRACT

This work studies aspects of RSA public key cryptography and its relationship with number theory, in particular with the properties of prime numbers. The methodology is based on bibliographical research. The study is based on the analysis of bibliographic texts. From analyzing the texts, the strengths of RSA public-key cryptography, also known as RSA asymmetric cryptography, are evident. As strengths, we can mention the greater security against attacks and code breakage by using big keys, even using super modern computers and with ultra-fast data processing and key management for all users of the system. The main disadvantage is the slowness of the encryption process, as it uses generally more complex algorithms and large keys.

Keywords: RSA cryptography. Prime numbers. Module *m* congruence.

SUMÁRIO

INTRODUÇÃO

Atualmente, as transações financeiras têm se tornado cada vez mais comuns no meio digital. Em 2019, segundo a Federação Brasileira de Bancos (Febraban) e a Deloitte, empresa de prestação de serviços, dentre eles, auditoria e assessoria financeira, 63% das transações foram efetuadas por canais digitais. O Brasil não só registrou aumento das transações em canais digitais de 3% em 2019 em relação a 2018 como também uma diminuição nos outros canais, como agências ou correspondentes, desde 2014. Ainda de acordo com a Febraban, em abril de 2020, 74% das transações foram feitas por pessoa físicas, um aumento de dez pontos percentuais comparado a janeiro do mesmo ano, também registrando queda das transações por outros canais de oito pontos percentuais, considerando o mesmo período (FEBRABAN, 2020).

Apesar dos números razoavelmente expressivos, poucos usuários compreendem os processos que envolvem a seguranças dos dados e das transações realizadas. Muitos apenas acreditam e confiam que estão seguros e sequer se dão conta da matemática que existe escondida em tudo isso ou da história que foi percorrida para que isso fosse possível hoje. Àqueles que conhecem os processos que possibilitam essa segurança, resta a preocupação de até que ponto estes métodos são seguros e o quanto podem confiar neles.

O presente trabalho trata de um dos métodos mais amplamente usados na segurança de informações compartilhadas pela internet, o sistema de criptografia de chave pública RSA, cujo acrônimo é formado pelas iniciais dos sobrenomes de seus criadores, Ronald Linn Rivest, Adi Shamir e Leonard Max Adleman, e que tem a teoria dos números, em especial, as propriedades dos números primos, como força basilar. Ademais, os detalhes da força que esse sistema tem e o importante papel que ele desempenha na segurança de informação moderna são discutidos nos capítulos seguintes.

1 CONCEITOS PRELIMINARES

Neste capítulo, são introduzidos alguns conceitos e resultados preliminares que serão úteis para o decorrer deste trabalho.

Definição 1.1. *(Divisor Comum)* Um divisor comum entre dois números inteiros a e b é um número inteiro que divide a e b simultaneamente.

Definição 1.2. *(Máximo Divisor Comum)* Máximo Divisor Comum ou MDC entre dois números inteiros a e b é o maior número k inteiro que divide a e b e tal que qualquer outro divisor de a e b é um divisor de k. Se k é o máximo divisor comum entre a e b, denotamos mdc $(a, b) = k$.

Definição 1.3. *(Produtório)* Produtório é a multiplicação de uma sequência de fatores que tem como resultado um produto. Dada uma sequência $\{k_i\}$, $i \in \mathbb{N}$, um produtório é definido por

$$\prod_{t=m}^{n} k_t = k_m \cdot k_{m+1} \cdot k_{m+2} \cdots k_{n-1} \cdot k_n.$$

Onde t é o índice do produtório, k_t é uma variável indexada que representa cada termo do produtório, m é o índice inicial e n é o índice final.

2 UM POUCO DE HISTÓRIA

Este trabalho se inicia com um pouco da história sobre os primórdios da teoria dos números, parte da matemática pura, e conta brevemente como alguns matemáticos, ao longo dos anos, fizeram importantes contribuições como demonstrações de teoremas, conjecturas, hipóteses e ferramentas que nos servem até hoje como base da criptografia RSA.

2.1 INFINITOS NÚMEROS PRIMOS

Um dos primeiros matemáticos dos quais se tem registro por ter trabalhado com os rudimentos da teoria dos números foi Euclides de Alexandria (350 a.C). Famoso por sua obra "Os Elementos" e conhecido como "pai da geometria", também fez grandes contribuições matemáticas com escritos sobre a teoria dos números nos seus primórdios (ROSEN, 2014).

Euclides percebeu que alguns números tinham propriedades especiais, isso é especialmente verdadeiro para os números primos. Na obra "Os Elementos", no livro VII, Euclides define número primo: "um número primo é o medido por uma unidade só", como apresenta Euclides (2009, p. 270).

Traduzindo a ideia contida na obra do matemático, que tem uma abordagem geométrica e, portanto, como medida e não puramente como o conceito de número que temos hoje, Euclides quer dizer que, número primo é aquele que só pode ser medido por ele mesmo e nenhuma outra medida inteira menor que ele, além da unidade, pode ser tomada para medir um primo. Na nossa linguagem seria o mesmo que dizer que um número é primo se só pode ser dividido por um (a unidade) ou por ele mesmo.

Mais que isso, Euclides quis mostrar que os números que tinham essa propriedade não eram, de forma alguma, finitos. Ele não só afirmou como provou sua afirmação. Assim nos deu uma demonstração de que esses números, os primos, são realmente infinitos.

Definição 2.1. *(Número Primo)* Um número inteiro p maior que 1 é dito primo se não existe d inteiro, com $1 < d < p$ tal que d divide p.

Definição 2.2. *(Número Composto)* Um número inteiro p maior que 1 é dito composto se existe d inteiro, com $1 < d < p$ tal que d divide p.

Pela definição, os números 0 e 1 não são primos nem compostos.

No livro IX, na proposição 20, Euclides (2009, p. 343) enuncia: "os números primos são mais numerosos do que toda a quantidade que tenha sido proposta de números primos", isso quer dizer que, dada uma quantidade finita de números primos, é sempre possível encontrar uma quantidade maior que a quantidade dada. Para facilitar o entendimento do texto, é possível reescrever a proposição e a demonstração que Euclides apresenta, da seguinte maneira:

Proposição 2.1. *(Infinitos Primos) Dado um conjunto com n números primos, existe um conjunto com* $n + 1$ *números primos:*

Prova. Seja, sem perda de generalidade, $A = \{a, b, c\}$ um conjunto com 3 elementos, 3 números primos. Seja d tal que $abc = d$. Dessa maneira d tem fatores a, b e c e, segundo as definições que Euclides apresenta no livro VII, é composto. Basta mostrar que sempre existe um conjunto de números primos maior (com mais elementos) que o conjunto dado.

Sendo $d = abc$, tome $x = d + 1$, logo $x \neq d$, e x pode ser primo ou não.

Se x é primo, então está demonstrada a proposição.

Se x não é primo, então algum fator p primo divide x.

Afirmação: esse fator p não é um elemento do conjunto A. Se a afirmação for provada verdadeira, então está provada a proposição.

De fato, se p pertencesse a A, então p dividiria d pois seria fator de d.

Mas se p divide d e divide $x = d + 1$ então p divide a diferença $x - d$.

Todavia, $x - d = 1$ e não existe primo que divida 1.

Logo, existe p primo tal que $p \notin A$ e existe um conjunto com 4 números primos distintos.

□

2.2 O PEQUENO TEOREMA DE FERMAT

Não foi apenas Euclides que despendeu esforços para entender as propriedades dos números primos, muitos outros matemáticos desempenharam papel de grande relevância para a construção do que hoje vemos como a teoria dos números.

Pierre de Fermat (1601–1665), foi outro matemático de grande importância para a teoria dos números e também responsável por nos apresentar a Geometria Analítica e ideias iniciais do cálculo infinitesimal (ROSEN, 2014).

Segundo Sautoy (SAUTOY, 2007), Fermat correspondia-se por cartas com seus colegas matemáticos e foi por meio dessas correspondências e dos registros marginais no livro "Aritmética", de Diofanto, que seu trabalho ficou conhecido e deu ainda mais suporte para a matemática. Dentre tantos trabalhos e anotações há um que merece destaque para o assunto da teoria do números – o Pequeno Teorema de Fermat.

Teorema 2.1. *(Pequeno Teorema de Fermat) Sejam* $a, p \in \mathbb{Z}$*, com* p *primo e* $\text{mdc}(a, p) = 1$*. Então*

$$a^{p-1} \equiv 1 (mod\ p).$$

O Pequeno Teorema de Fermat também foi demonstrado mais tarde por Leonhard Euler (1707–1783). A demonstração aqui apresentada utilizará apenas as relações de congruência, operações módulo m e fatorial.

Antes de seguir a demonstração deste teorema, é importante compreender as ferramentas usadas nela.

Definição 2.3. *(Coprimos)* Chamamos de coprimos ou primos entre si, os números para os quais o único divisor comum é 1.

Teorema 2.2. *(Algoritmo da Divisão Euclidiana) Dados $a, b \in \mathbb{Z}$, com $b \neq 0$, existem únicos inteiros q e r, chamados respectivamente de quociente e resto, tais que*

$$a = bq + r, \quad com\ 0 \leq r < |b|.$$

A demonstração será omitida mas pode ser encontrada em (CASTRO, 2010).

Definição 2.4. *(Relação de Congruência Módulo m)* Sejam $a, b, m \in \mathbb{Z}$, com $m \neq 0$. Dizemos que a é congruente a b módulo m, e denotamos por $a \equiv b \pmod{m}$, se m divide a diferença $(a - b)$, o que denotamos por $m \mid (a - b)$. Se m não divide $(a - b)$, o que denotamos por $m \nmid (a - b)$, dizemos que a é incongruente a b módulo m e denotamos por $a \not\equiv b \pmod{m}$.

Pela definição de congruência módulo m fica implícita a relação com a divisão euclidiana, uma vez que $a \equiv b \pmod{m}$ significa que $m \mid (a - b)$, donde $a - b = km$ para algum $k \in \mathbb{Z}$, o que, reescrevendo, fica $a = km + b$.

Mas, no caso da congruência, b não é, necessariamente o resto da divisão euclidiana de a por m, mas mostra que a e b têm o mesmo resto quando divididos por m.

Proposição 2.2. *Sejam $a, b, c, m \in \mathbb{Z}$, com $m > 0$. Valem as seguintes propriedades:*

a) Reflexividade: $a \equiv a \pmod{m}$.

b) Simetria: $a \equiv b \pmod{m}$ então $b \equiv a \pmod{m}$.

c) Transitividade: Se $a \equiv b \pmod{m}$ e $b \equiv c \pmod{m}$, então $a \equiv c \pmod{m}$.

Prova. a) $m \mid (a - a)$, pois $(a - a) = 0$.

b) Se $a \equiv b \pmod{m}$, então $m \mid (a - b)$. Assim $m \mid -(a - b)$, logo $m \mid (b - a)$ e, por definição, $b \equiv a \pmod{m}$.

c) Se $a \equiv b \pmod{m}$ e $b \equiv c \pmod{m}$, então $m \mid (a - b)$ e $m \mid (b - c)$. Assim $m \mid (a - b) + (b - c)$, logo $m \mid (a - c)$ e, por definição, $a \equiv c \pmod{m}$.

□

Proposição 2.3. *Sejam $a, b, a_1, a_2, b_1, b_2, m \in \mathbb{Z}$, com $m > 0$. Se $a_1 \equiv b_1 \pmod{m}$ e $a_2 \equiv b_2 \pmod{m}$, ou se $a \equiv b \pmod{m}$, valem as seguintes propriedades:*

a) $a + k \equiv b + k \pmod{m}$, $\forall k \in \mathbb{Z}$.

b) $ak \equiv bk \pmod{m}$, $\forall k \in \mathbb{Z}$.

c) $a_1 + a_2 \equiv b_1 + b_2 \pmod{m}$.

d) $a_1 - a_2 \equiv b_1 - b_2$ *(mod m).*

e) $a_1 a_2 \equiv b_1 b_2$ *(mod m).*

f) Se $a + k \equiv b + k$ *(mod m)* $\forall k \in \mathbb{Z}$*, então* $a \equiv b$ *(mod m).*

g) Se $ak \equiv bk$ *(mod m) e k é coprimo com m, então* $a \equiv b$ *(mod m).*

Prova. a) Se $a \equiv b \pmod{m}$, então $m \mid (a - b)$. Da identidade $(a + k) - (b + k) = a - b$, temos que $m \mid [(a + k) - (b + k)]$. Portanto, $a + k \equiv b + k \pmod{m}$, $\forall k \in \mathbb{Z}$.

b) Se $a \equiv b \pmod{m}$, então $m \mid (a - b)$. Segue que $m \mid k(a - b)$, ou seja, $m|(ak - bk)$. Portanto, $ak \equiv bk \pmod{m}$.

c) Como $(a_1 + a_2) - (b_1 + b_2) = (a_1 - b_1) + (a_2 - b_2)$, temos que $(a_1 + a_2) - (b_1 + b_2) = km - lm = (k - l)m$, para $k, l \in \mathbb{Z}$ tais que $km = a_1 - b_1$ e $lm = a_2 - b_2$. Portanto, $m \mid [(a_1 + a_2) - (b_1 + b_2)]$, ou seja, $a_1 + a_2 \equiv b_1 + b_2 \pmod{m}$.

d) Como $(a_1 - a_2) - (b_1 - b_2) = (a_1 - b_1) - (a_2 - b_2)$, temos que $(a_1 - a_2) - (b_1 - b_2) = km - lm = (k + l)m$, para $k, l \in \mathbb{Z}$ tais que $km = a_1 - a_2$ e $lm = b_1 - b_2$. Portanto, $m \mid [(a_1 - a_2) - (b_1 - b_2)]$, ou seja, $a_1 - a_2 \equiv b_1 - b_2 \pmod{m}$.

e) Como $a_1 a_2 - b_1 b_2 = a_1 a_2 - b_1 a_2 + b_1 a_2 - b_1 b_2 = a_2(a_1 - b_1) + b_1(a_2) - b_2)$, temos que $a_1 a_2 - b_1 b_2 = a_2(km) + b_1(lm) = m(a_2 k + b_1 l)$, para $k, l \in \mathbb{Z}$ tais que $km = a_1 - a_2$ e $lm = b_1 - b_2$. Portanto, $m|(a_1 a_2 - b_1 b_2)$, ou seja, $a_1 a_2 \equiv b_1 b_2 \pmod{m}$.

f) Se $a + k \equiv b + k \pmod{m}$, então $m \mid (a + k) - (b + k)$. Como $(a + k) - (b + k) = a + k - b - k = a - b$, segue que $m \mid (a - b)$, donde $a \equiv b \pmod{m}$.

g) Sejam $ak \equiv bk \pmod{m}$ e $d = \text{mdc}(k, m) = 1$. De $ak \equiv bk \pmod{m}$, segue que existe $x \in \mathbb{Z}$ tal que $k(a - b) = xm$. Desse modo $m \mid k \cdot (a - b)$. Por hipótese, $\text{mdc}(k, m) = 1$, então $m \mid (a - b)$, ou seja $a \equiv b \pmod{m}$.

□

Apresentadas as ferramentas usadas, passemos à demonstração do Pequeno Teorema de Fermat.

Enunciado: Sejam $a, p \in \mathbb{Z}$, com p primo e $\text{mdc}(a, p) = 1$. Então

$$a^{p-1} \equiv 1 (mod\ p).$$

Prova. Sejam $A = \{1, 2, 3, ..., (p - 1)\}$ e $B = \{1a, 2a, 3a, ..., (p - 1)a\}$ conjuntos.

De $\text{mdc}(a, p) = 1$ implica que $p \nmid a$, logo p não divide nenhum elemento de B.

De fato, pois, por construção do conjunto B, os coeficientes de a são menores que p.

E por p primo, os coeficientes de a não são fatores de p.

POr conseguinte, $y \not\equiv 0 \pmod{p}$, para todo $y \in B$.

Sejam $r, s \in A$ tais que $ra \equiv sa \pmod{p}$.

Pelas propriedades 2 e 7 da proposição 2.7, tem-se que $r \equiv s \pmod{p}$.

Como, por hipótese, $\text{mdc}(a, p) = 1$, resta que $r = s$.

De fato, pois todo elemento de A é menor que p e, na divisão por p, cada um deixa um resto diferente do outro, além disso $r \equiv s \pmod{p}$ se r e s deixam mesmo resto quando divididos por p, o que implica $r = s$.

Logo, os elementos de B são incongruentes entre si.

Mas os elementos de A são congruentes, de algum modo, com os respectivos elementos de B.

Assim, pela propriedade 5:

$$(p-1)! = 1 \cdot 2 \cdot 3 \cdots (p-1) \equiv 1a \cdot 2a \cdot 3a \cdots (p-1)a.$$
$$\Rightarrow (p-1)! \equiv a^{p-1} \cdot (p-1)! \ (mod\ p).$$

De p primo, tem-se que $\text{mdc}(p, (p-1)!) = 1$.

De p primo, $p \geq 2$, logo $(p-1) \geq 1$ e $(p-1)! \neq 0$ por definição de fatorial.

Logo, podemos cancelar $(p-1)!$ na congruência $(p-1)! \equiv a^{p-1} \cdot (p-1)! \ (mod\ p)$, obtendo

$$a^{p-1} \equiv 1 \ (mod\ p).$$

□

2.3 O TEOREMA DE EULER E A FUNÇÃO φ DE EULER

Além da demonstração do Pequeno Teorema de Fermat, Euler também nos deu uma generalização desse teorema.

O Pequeno Teorema de Fermat vale para quando p é um número primo, mas Euler conseguiu generalizá-lo para quando substituirmos p por um número composto, todavia foi necessária uma modificação restritiva para a escolha desse número.

A generalização de Euler ficou conhecida como Teorema de Euler. Vale indicar que é o Teorema de Euler para a teoria dos números, já que Euler deixou sua marca em várias áreas da matemática.

A modificação para a generalização está essencialmente no expoente do número a, que agora é denotado por $\varphi(n)$, na qual n é um inteiro coprimo de a, que não precisa ser necessariamente primo. Para entender melhor a generalização, é necessário entender o que significa a expressão $\varphi(n)$, que indica a Função Totiente ou Função φ de Euler.

Definição 2.5. *(Função φ de Euler)* Seja Seja $n \in \mathbb{N}$. Definimos $\varphi(n)$ como a quantidade de números menores ou iguais a n que são coprimos com n.

$$\varphi(n) = \#\{x \in \mathbb{N} \mid x \leq n \ \wedge \ \text{mdc}(n, x) = 1\}.$$

Por exemplo, $\varphi(15) = 8$, pois $1, 2, 4, 7, 8, 11, 13, 14$ são todos coprimos com 15 e não maiores que 15. É de fácil e imediata verificação, pela definição de $\varphi(n)$, que, se n for primo, então $\varphi(n) = (n - 1)$, além de que $\varphi(1) = 1$, pois $1 \leq 1$ e $\text{mdc}(1, 1) = 1$.

Proposição 2.4. *Sejam $a, p, x, y, k \in \mathbb{N}^*$, com p primo e $\text{mdc}(x, y) = 1$. Valem as seguintes propriedades:*

a) $\varphi(p^k) = p^k - p^{k-1}$.

b) $\varphi(xy) = \varphi(x)\varphi(y)$.

c) Se $a = p_1^{k_1} p_2^{k_2} \cdots p_n^{k_n}$, com $p_1^{k_1} p_2^{k_2} \cdots p_n^{k_n}$ fatores primos de a, então

$$\varphi(a) = a\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_n}\right).$$

.

Prova. a) Precisamos provar que a quantidade de números menores ou iguais a p^k e que são coprimos com p^k é exatamente $p^k - p^{k-1}$.

Um número não é coprimo com p^k se, e somente se, ele é um múltiplo de p.

Existem exatamente $\dfrac{p^k}{p} = p^{(k-1)}$ números entre 1 e p^k que são divisíveis por p.

Então basta subtrair os múltiplos de p e teremos que $\varphi(p^k) = p^k - p^{(k-1)}$.

b) Consideremos os números de 1 até xy dispostos em uma tabela, em sequência, em x linhas e y colunas.

Dessa maneira existem exatamente $\varphi(xy)$ números coprimos com xy na tabela.

Por hipótese $\text{mdc}(x, y) = 1$, por consequência $\text{mdc}(a, xy) = 1 \Leftrightarrow \text{mdc}(a, x) = 1$ e $\text{mdc}(a, y) = 1$, simultaneamente.

Logo, se $1 \leqslant i \leqslant y$ é tal que $\text{mdc}(i, y) = c > 1$, então todos os elementos da i-ésima coluna não são coprimos com y, pois $\text{mdc}(i, ki + y) = \text{mdc}(i, y)$, que não são coprimos com xy.

Portanto os números que são coprimos com xy estão nas $\varphi(y)$ colunas cujos primeiros elementos são coprimos com xy.

Tomemos então a i-ésima coluna tal que $\text{mdc}(i, y) = 1$. Os elementos dessa coluna são da forma $(ky + i)$, com $0 \leqslant k \leqslant x - 1$. Nessas condições, a coluna apresenta x números que deixam restos distintos na divisão por x e temos, entre eles, $\varphi(x)$ números coprimos com x

Logo, em cada uma das $\varphi(y)$ colunas temos $\varphi(x)$ coprimos com x e com y.

Concluímos que $\varphi(xy) = \varphi(x)\varphi(y)$.

c) Pela propriedade 1 temos que

$$\varphi(p^k) = p^k - p^{k-1} = p^k\left(1 - \frac{1}{p}\right).$$

Então

$$\begin{aligned}\varphi(a) = \varphi(p_1^{k_1}p_2^{k_2}\cdots p_n^{k_n}) &= p_1^{k_1}\left(1-\frac{1}{p_1}\right)p_2^{k_2}\left(1-\frac{1}{p_2}\right)\cdots p_n^{k_n}\left(1-\frac{1}{p_n}\right)\\ &= p_1^{k_1}p_2^{k_2}\cdots p_n^{k_n}\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_n}\right)\\ &= a\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_n}\right).\end{aligned}$$

□

O Teorema de Euler estabelece o seguinte:

Teorema 2.3. *(Teorema de Euler) Sejam* $a, n \in \mathbb{Z}$*, com* $\text{mdc}(a, n) = 1$*. Então*

$$a^{\varphi(n)} \equiv 1 \ (mod\ n).$$

Prova. Sejam $A = \{r_1, r_2, ...r_{\varphi(n)}\}$ o conjunto dos restos não nulos na divisão por n, isto é, $\text{mdc}(r_i, n) = 1$, para todo $i = 1, 2, ..., \varphi(n)$ e $B = \{ar_1, ar_2, ..., ar_{\varphi(n)}\}$.

Por hipótese $\text{mdc}(a, n) = 1$, segue que $\text{mdc}(ar_i, n) = 1$.

Por construção, os elementos de A são congruentes, de algum modo, com os elementos de B. Então

$$\begin{aligned}ar_1ar_2\cdots ar_{\varphi(n)} &\equiv r_1r_2\cdots r_{\varphi(n)}\\ &\Rightarrow a^{\varphi(n)}r_1r_2\cdots r_{\varphi(n)} \equiv r_1r_2\cdots r_{\varphi(n)}\\ &\Rightarrow a^{\varphi(n)} \equiv 1 \ (mod\ n).\end{aligned}$$

□

Se não sabemos se um determinado número é primo, então calcular $\varphi(n)$ pode ser difícil, mas se soubermos, fica muito mais fácil verificar os coprimos.

Euler sabia disso e mostrou como calcular $\varphi(n)$, mas para isso, teríamos que saber a fatoração do número dado, o que ainda consistia em problema muito difícil, pois até hoje os algoritmos existentes para isso ainda são extremamente ineficientes para números grandes, quanto mais naquela época. Para calcular $\varphi(n)$ conhecendo-se a fatoração, procede-se da seguinte maneira:

Se n é primo já vimos que $\varphi(n) = n - 1$ e vale o Pequeno Teorema de Fermat.

Se n é composto, então, pelo Teorema Fundamental da Aritmética, $n = p_1^{k_1} \cdots p_s^{k_s}$ onde os p_i são fatores primos distintos de n e os k_i são a multiplicidade dos respectivos fatores, com $i \in \{1, 2, ..., s\}$. Então

$$\varphi(n) = (p-1)p_1^{k_1-1} \cdots (p_s^{k_s-1}) = n \prod_{p|n} \left(1 - \frac{1}{p}\right).$$

A notação $p \mid n$ sob o produtório representa que são usados os primos p que são fatores de n.

2.4 ESTIMATIVA DA QUANTIDADE DE PRIMOS

Mas enquanto tudo isso acontecia, a busca por uma pista da ordenação dos números primos continuava sem descanso. Outro matemático que fez grandes descobertas e que fez surgir os rudimentos da técnica utilizada neste trabalho para demonstrar o Pequeno Teorema de Fermat, a operação com congruência em módulo, foi Carl Friedrich Gauss (1777–1855).

Gauss, como outros matemáticos na busca de respostas, foi responsável pela criação de mais áreas da matemática, como a Teoria dos Números Algébricos. Na tentativa de encontrar uma resposta sobre a questão da distribuição dos números primos, por mais que procurasse, não conseguia determinar quanto deveria se contar para encontrar um número primo a partir de um outro dado (SAUTOY, 2007). Afinal, seriam os primos distribuídos segundo uma aleatoriedade da natureza?

Apesar de não conseguir encontrar a ordenação dos números primos, Gauss acreditava que tinha encontrado uma forma de “contar” os números primos dado um intervalo de números. Acreditava que essa forma era, no mínimo, uma aproximação muito boa da quantidade de primos que se deve esperar encontrar dado um intervalo entre os números.

Se alguém perguntasse, por exemplo, quantos primos existem entre os números 1 e 10.000, Gauss acreditava que sua fórmula poderia estimar com razoável precisão a quantidade de primos existentes nesse intervalo.

Mesmo não conseguindo determinar onde o próximo número primo surgiria, dado um número primo anterior, prever a quantidade possível de primos num dado intervalo poderia ser a luz para saber quanto de trabalho se teria para encontrar os próximos primos. Por exemplo, supondo que a estratégia desenvolvida por Gauss estivesse correta, se em dado intervalo tivéssemos apenas dez números primos a serem encontrados, não seria necessário continuar a procurar mais primos após encontrar o décimo número primo, afinal, não existiria nenhum outro nesse intervalo.

O problema era que a fórmula que Gauss desenvolveu era uma aproximação, não uma garantia que existiriam exatamente a quantidade determinada pela fórmula, poderia variar para mais ou para menos, mas não muito.

A descoberta de Gauss se deu de uma relação íntima que ele percebeu entre os números primos e os logarítmos naturais. A fórmula de Gauss estimava que a quantidade de primos entre

1 e n, com n um número natural maior que 1 era, aproximadamente,

$$\frac{n}{\ln(n)}.$$

Essa proximação subestimava o número real de primos. A quantidade real de números primos era maior que a estimativa de Gauss e à medida que n se tornasse maior a estimativa de Gauss também se afastaria do número veradeiro de primos, mas ainda assim serviria como um limitante inferior.

Depois de alguns anos, Gauss conseguiu refinar um pouco mais sua fórmula e chegar a uma precisão um pouco maior sobre a quantidade de primos estimada dado um intervalo, todavia ainda não podia afirmar exatamente a quantidade de primos, era ainda uma estimativa, mas um pouco melhor e precisa que a anterior. A nova fórmula foi nomeada $Li(n)$ e a nova estimativa, ao contrário da anterior, superestimava a quantidade de números primos.

$Li(n)$ é dada pela integral

$$\int_2^n \frac{dt}{\ln(t)}.$$

Por exemplo, a diferença entre a quantidade real de numeros primos e $\frac{n}{\ln(n)}$, para $n = 10^8$ é de 332744, isto é, a primeira estimativa de Gauss subestimava a quantidade real em 332744, enquanto a fórmula aperfeiçoada superestimava a quantidade real em apenas 753.

Se o número considerado fosse ainda maior, por exemplo $n = 10^{10}$, então teríamos a diferença entre o número real de primos e $\frac{n}{\ln(n)}$ de 20758030, enquanto para $Li(n)$ teríamos a diferença de 3103.

A matemática parecia cada vez mais próxima de responder a pergunta que deixava os matemáticos inquietos: é possível identificar uma ordenação, um padrão, na ocorrência dos números primos? Afinal, já se conhecia uma aproximação da quantidade de primos dado um intervalo, conhecia-se também uma maneira de se contar os coprimos menores ou iguais a um dado número, mesmo que não muito eficiente.

Contudo, todo esse trabalho ainda não era suficiente, todas as ferramentas desenvolvidas ainda não tinham a força necessária para quebrar a barreira que escondia o segredo dos números primos.

2.5 A QUEBRA DA CIFRA DA MÁQUINA ENIGMA

No século, XIX, durante a Segunda Guerra Mundial, enquanto matemáticos ainda se preocupavam com problemas como a previsibilidade da ordenação dos primos, outra pessoa começa a dar passos importantes para a história da criptografia. Alan Mathison Turing (1912–1954), como muitos outros matemáticos antes dele, não era apenas um matemático, Turing também era criptoanalista e biólogo teórico.

O trabalho de Turing foi responsável por quebrar as cifras alemãs e assim conseguir ler as mensagens cifradas da força naval alemã e antecipar seus movimentos, uma contribuição que salvou muitas vidas (SAUTOY, 2007).

A inteligência alemã usava uma máquina, chamada Enigma, muito parecida esteticamente com uma máquina de escrever, de funcionamento eletromecânico, isto é, tanto elétrico quanto mecânico, tanto para criptografar quanto para descriptografar uma mensagem, e assim deixá-la ininteligível para quem não tivesse a máquina Enigma.

A função da máquina era transformar um texto inteligível em um texto ou código não inteligível, uma substituição de caracteres de forma lógica, muito mais sofisticado que uma substituição simples, como mostra Shokranian (SHOKRANIAN, 2005) sobre a Cifra de César, um sistema simples de substituição de de posição das letras do alfabeto usado.

A máquina Enigma funciona com rotores que, a cada tecla pressionada, giram e assumem uma outra posição, embaralhando as letras na hora da visualização da mensagem que seria transmitida por rádio posteriormente. Por conseguinte, uma configuração inicial diferente implica numa mensagem criptografada diferente e, na leitura, de uma mensagem diferente da original. Por isso, apenas possuir uma máquina Enigma não seria suficiente para compreender a mensagem, pois mesmo com uma máquina dessas seria quase impossível decifrar uma mensagem sem conhecer a posição inicial do rotor da máquina.

Acreditando que o sistema da Enigma era razoavelmente forte, os alemães estavam bastante confiantes de que os adversários não conseguiriam decifrar suas mensagens, mesmo que interceptassem as mensagens de rádio, ou até mesmo se conseguissem uma das máquinas Enigma, porque teriam ainda que saber quais configurações dos rotores deveriam ser usadas para decifrar corretamente a mensagem (SINGH, 1999).

Todavia, o sistema da máquina Enigma não era tão forte assim, em questão de segurança, como imaginavam os alemães, além disso, para que se tivesse um uso adequado da máquina, era necessário o compartilhamento das mesmas chaves, que eram as configurações iniciais dos rotores, o que pode causar um problema de logística e segurança que não podia ser ignorado.

Os alemães precisavam compartilhar de maneira secreta qual a posição inicial dos rotores usados na codificação da mensagem com os operadores das máquinas responsáveis por decodificar a mensagem, somente assim garantiriam que a mensagem recebida poderia ser lida.

A partir do sucesso de Turing em quebrar a segurança da máquina Enigma, certamente a prudência na segurança das informações deveria ser aumentada. Uma mensagem secreta deveria ser indecifrável para aqueles que não são seus destinatários mesmo que a interceptassem ou, no mínimo, apresentar uma dificuldade muito grande de decifração a ponto de não ser viável o esforço para ler a mensagem secreta ou de demorar tempo o suficiente para que quando a mensagem pudesse ser lida por outra pessoa que não seja o destinatário, já não seja, de forma alguma, relevante.

A necessidade de aumentar a segurança nas mensagens secretas se estendeu por mais alguns anos e instigou alguns avanços importantes na área da segurança da informação.

3 GERANDO E TESTANDO NÚMEROS PRIMOS

É importante, antes de apresentar a Criptografia RSA, mostrar os avanços que a matemática apresentou nas técnicas para gerar números primos e verificar primalidade, bem como suas dificuldades e limitações. Assim é possível entender a relação da Criptografia RSA, um sistema relativamente moderno, criado na segunda metade do século XX, mas que se apoia em descobertas matemáticas que surgiram há muito tempo e em um problema sobre os números primos que até hoje ocupa vários matemáticos, a fatoração de números grandes..

Neste capítulo são apresentados os números de Mersenne, números de Fermat, números de Chermichael e suas características, além de algoritmos para encontrar números primos

3.1 CRIVO DE ERATÓSTENES

Muitos matemáticos, desde a antiguidade, desejaram criar um método para gerar números primos. Afinal, isso também poderia ser um caminho para descobrir a ordenação desses números tão misteriosos.

O matemático grego Eratóstenes de Cirene (276 a.C.–194 a.C.) criou um método que selecionava números primos. Era um método bastante rudimentar, mas que funcionava.

O método desenvolvido por Eratóstenes se baseava em tomar uma quantidade de números inteiros positivos e eliminar os que eram compostos, a partir do menor desses números que fosse maior que 1. Escolhendo o menor número maior que 1, elimina-se todos os múltiplos dele. Após a primeira etapa de eliminação de múltiplos, procede-se semelhantemente, escolhendo o próximo número dentre os restantes que seja imediatamente maior que o número escolhido anteriormente e eliminando os múltiplos deste. Repete-se o processo até que não reste mais números a serem eliminados. Ao final restariam apenas números primos. Por exemplo, tomemos os números de 2 a 30 e apliquemos o Crivo de Eratóstenes:

$$2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30.$$

E então tomamos o número 2, que é o menor desses números que é maior que 1 e eliminamos todos seus múltiplos da lista:

$$2,\ 3,\ \cancel{4},\ 5,\ \cancel{6},\ 7,\ \cancel{8},\ 9,\ \cancel{10},\ 11,\ \cancel{12},\ 13,\ \cancel{14},\ 15,\ \cancel{16},\ 17,\ \cancel{18},\ 19,\ \cancel{20},\ 21,\ \cancel{22},\ 23,\ \cancel{24},\ 25,\ \cancel{26},\ 27,\ \cancel{28},\ 29,\ \cancel{30}.$$

Rearranjando os números, excluindo aqueles que já foram cancelados:

$$2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29,.$$

Tomamos agora o número 3, que é o segundo menor número, e cancelamos todos os seus múltiplos:

$$2, 3, 5, 7, \cancel{9}, 11, 13, \cancel{15}, 17, 19, \cancel{21}, 23, 25, \cancel{27}, 29.$$

Rearranjando os números, excluindo aqueles novos cancelados :

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 25, 29.$$

Tomamos o terceiro menor número e cancelamos todos os seus múltiplos:

$$2, 3, 5, 7, 11, 13, 17, 19, 23, \cancel{25}, 29.$$

Rearranjando os números, excluindo aqueles novos cancelados:

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29.$$

No exemplo apresentado, já acabaram os números compostos, visto que se testarmos todos os outros restantes não encontraremos nenhum múltiplo deles na lista. Todos os números que restaram são primos.

O Crivo de Eratóstenes encontra os primos dessa maneira. O método é um tanto quanto simples e demorado, mas funciona. O problema é a demora para se verificar quantos dos números dados são primos, principalmente se a quantidade de números dados for muito grande.

3.2 NÚMEROS DE MERSENNE E NÚMEROS DE FERMAT

Marin Mersenne (1588–1648) foi um frade e matemático amador que se correspondia frequentemente com Fermat e muitos outros matemáticos bastante influentes da sua época. Também interessado na questão da ordenação dos números primos (SAUTOY, 2007), os estudos de Mersenne resultaram em um caminho primário para tentar gerar números primos. O método que Mersenne desenvolveu parecia bastante promissor inicialmente.

Primeiramente Mersenne conjecturou que os números da forma $2^n - 1$, com n um inteiro maior que 1, seriam primos, mas logo percebeu que nem todos os números que fossem substituídos por n nessa expressão geravam números primos. Mersenne conseguiu demonstrar que se n fosse composto, então $2^n - 1$ também seria composto.

Usaremos a notação $M(n) = 2^n - 1$ para designar o Número de Mersenne gerado por n.

Proposição 3.1. *Se n é um número inteiro é composto maio que* 1*, então* $2^n - 1$ *é composto.*

Prova. Seja $n = ab$ composto, $a > 1$ e $b > 1$. Então

$$2^n - 1 = 2^{ab} - 1 = (2^a - 1)(2^{a(b-1)} + 2^{a(b-2)} + \ ... \ + 2^a + 1).$$

É fácil ver, depois que a expressão está aberta, que $(2^a - 1)$ é fator de $2^n - 1 = M(n)$. Reescrevendo $(2^a - 1) = M(a)$, segue que $M(a)|M(n)$, o que mostra que $M(n)$ é composto. □

Mas, então, se não vale para os n compostos, deve valer para os n primos, certo? Curiosamente a resposta é não. Os Números de Mersenne são sempre compostos se n é composto e somente se n é primo é que esses números têm a chance de serem primos, mas ainda assim não são garantidamente primos. Por exemplo, para o número 11, que é primo, teríamos

$$M(11) = 2^{11} - 1 = 2048 - 1 = 2047 = 23 \cdot 89.$$

Portanto, o método de Mersenne não garante um número primo sempre que n for primo, mas n primo é condição necessária para que $M(n)$ o seja. Mesmo nessas condições o método dá uma ideia de por onde pode-se procurar, apesar de não ser eficiente.

Fermat também foi capaz de desenvolver um método em busca de gerar primos. Entendendo o que fez Mersenne, Fermat acreditou que os números primos poderiam ser gerados de uma maneira semelhante. Propôs que os números da forma $2^{2^n} + 1$ eram primos, para valores inteiros não negativos e testou para os valores de n entre 0 e 6.

Chamaremos de $F(n) = 2^{2^n} + 1$ os Números de Fermat gerados por n. Calculando os valores de $F(n)$, para os n entres 0 e 6 temos $F(0) = 3$ e $F(1) = 5$ ambos com 1 algarismo, $F(2) = 17$ com 2 algarismos, $F(3) = 257$ com 3 algarismos, $F(4) = 65537$ com 5 algarismos, $F(5) = 4294967297$ com 10 algarismos e $F(6) = 18446744073709551617$ que tem 20 algarismos
.

Percebe-se que a quantidade de algarismos cresce muito rapidamente à medida que n cresce em uma unidade. O que Fermat não percebeu é que $F(5)$ é, na verdade, um número composto, o que foi verificado depois por Euler, e até hoje não se conhece Números de Fermat primos com $n > 5$, assim $F(6)$ também não é primo.

Como outros Números de Mersenne primos já foram encontrados, pode-se dizer que o método de Mersenne é mais eficiente que o de Fermat. O caso é que tanto Fermat quanto Mersenne não conseguiram encontrar um método de produzir primos, vez ou outra seus metódos falhariam.

Com as tentativas de criação de um método que gerasse números primos, naturalmente deveria aparecer algum outro que verificasse e certificasse a primalidade dos números gerados, até mesmo para poder validar o método que os gerou. Com isso em mente, alguns testes para determinar a primalidade de números começaram a ser criados.

3.3 O TESTE DE LEIBNIZ E OS NÚMEROS DE CARMICHAEL

Poucos anos após Mersenne e Fermat criarem seus métodos, Gottfried Wilhelm Leibniz (1646–1716), matemático e filósofo, usando do conhecimento do Pequeno Teorema de Fermat acreditava ter desenvolvido um teste de primalidade eficiente.

Recordemos que o Pequeno Teorema de Fermat diz que se $a, p \in \mathbb{Z}$, com p primo e $\text{mdc}(a, p) = 1$, então $a^{p-1} \equiv 1 (mod\ p)$, ou equivalentemente, se $a, p \in \mathbb{Z}$, com p primo e

mdc(a, p) = 1, então $a^p \equiv a(mod\ p)$.

Como resultado desse teorema, se tivéssemos $a, n \in \mathbb{Z}$, com n um número inteiro ímpar qualquer e mdc(a, n) = 1, então se $a^n \not\equiv a(mod\ n)$ poderíamos concluir que n não é primo, pois contradiz o teorema. Isso nos mostra indícios de um caminho para verificar não a primalidade de números mas se eles são compostos.

Por se tratar de uma operação com congruência módulo n, pelas propriedades de congruência, verificamos que todo número inteiro x é congruente a algum um outro inteiro y tal que $0 \leqslant y \leqslant n-1$, então só precisamos verificar os números a tais que $1 < a < n-1$. Os números 0, 1 e $(n-1)$ não estão incluídos porque a equação do Pequeno Teorema de Fermat, $a^n \equiv a(mod\ n)$, é sempre satisfeita se $a \in A = \{0, 1, (n-1)\}$

Para $a = 0$ e $a = 1$ é imediato, e para $a = n-1$ temos $a^n \equiv a \pmod{n}$, pois se $n \geq 3$ é ímpar, $n - 1 \equiv -1 \pmod{n} \Rightarrow (n-1)^n \equiv (-1)^n = -1 \equiv n - 1 \pmod{n}$.

A ideia de Leibniz nos mostra uma forma de identificar se um número é composto sem que para isso seja necessário calcular os seus fatores.

Leibniz acreditava que se um dado número n ímpar satisfizesse a equação $a^{n-1} \equiv 1 \pmod{n}$, para algum a tal que $1 < a < n-1$, então n seria primo. Infelizmente Leibniz estava equivocado.

O Pequeno Teorema de Fermat nos traz a informação de que dados $a, p \in \mathbb{Z}$, com p primo e mdc(a, p) = 1, então $a^p \equiv a(mod\ p)$, mas não nos garante a recíproca, isto é, dados $a, p \in \mathbb{Z}$, se $a^p \equiv a(mod\ p)$, então p é primo. Um contra exemplo é $n = 341$, pois

$$2^{(341-1)} = 2^{340} \equiv 1 \pmod{341}.$$

Pelo critério de Leibniz, 341 seria primo, todavia $341 = 31 \cdot 11$, logo, é composto, apontando a falha no critério de Leibniz.

Os números que passam pelo critério de Leibniz, mas na verdade são compostos, são conhecidos como pseudoprimos e, no exemplo apresentado, 341 é um pseudoprimo para a base 2, já que usamos $a = 2$. Vale lembrar que Leibniz tomava $a = 2$ para sua fórumla, que é a base mais simples de calcular.

Parece que não há nenhuma novidade no teste de Leibniz, afinal todos os métodos anteriores, e até mesmo o dele, não conseguiram gerar primos, mas um passo significativo foi dado com esse último trabalho.

Assim como já vimos, seria necessário testar apenas os números a tais que $1 < a < n-1$ para verificar se o número n é composto ou não. Leibniz usou apenas $a = 2$, mas se usasse $a = 3$ encontraria

$$3^{(341-1)} = 3^{340} \equiv 56 \pmod{341},$$

que indica imediatamente que 341 não é primo.

Com um sistema de computação algébrica, não é difícil perceber que entre 1 e 10^9, existem 50847534 números primos, mas existem 5597 pseudoprimos para a base 2. No entanto,

para a base 2 e 3 simultaneamente, apenas 1272 pseudoprimos. Isso nos mostra que, apesar do teste de Leibniz não ser muito preciso testando apenas uma base, é possível aumentar a precisão testando mais bases (COUTINHO, 2014).

Em resumo, para números relativamente pequenos, o teste de Leibniz é razoável, mas à medida que os números ficam maiores, é necessário testar muito mais bases para ter certeza se um número é composto ou não.

3.4 O TESTE DE MILLER-RABIN

Recentemente, em 1976, Gary Lee Miller e Michael Oser Rabin (1931–), propuseram um novo teste de primalidade melhorando razoavelmente o teste de Leibniz. O teste de Miller-Rabin é um pouco melhor porque consegue identificar até os pseudoprimos que o teste de Leibniz não conseguia sem testar todas as bases.

Para aplicar o teste de Miller-Rabin algumas condições precisam ser satisfeitas. Sejam $n \in \mathbb{Z}$, $n > 0$, um número ímpar e $a \in \mathbb{Z}$, com $1 < a < n-1$. Como $n-1$ é par, podemos reescrever convenientemente $n - 1 = 2^k q$, com $q, k \in \mathbb{Z}$ tais que q é ímpar e $k \geqslant 1$, sendo k a maior potência de 2 que satisfaz a igualdade.

O teste de Miller-Rabin busca verificar se alguma das potências

$$a^q,\ a^{2q},\ ...,\ a^{2^{k-1}q},\ a^{2^k q}$$

é congruente a 1 (mod n). De fato, se n for primo, pelo menos uma dessas potências é congruente a 1 (mod n), esse resultado já é conhecido do Pequeno Teorema de Fermat.

Se considerarmos x o menor expoente que satisfaça a expressão $a^{2^x q} \equiv 1$ (mod n), se $x \geqslant 1$, é possível reescrever a expressão da seguinte forma

$$a^{2^x q} - 1 = (a^{2^{x-1}q} - 1)(a^{2^{x-1}q} + 1).$$

Como n é primo por hipótese e divide $a^{2^x q} - 1$, segue que n divide o produto $(a^{2^{x-1}q} - 1)(a^{2^{x-1}q+1})$. Logo, n divide $(a^{2^{x-1}q} - 1)$ ou divide $(a^{2^{x-1}q} + 1)$. Por hipótese x é o menor expoente tal que n divide $a^{2^x q} - 1$ e $x - 1 < x$, segue que $(a^{2^{x-1}q} - 1)$ não é divisível por x. Então $(a^{2^{x-1}q} + 1)$ é divisível por x. Concluímos que $(a^{2^{x-1}q}) \equiv -1$ (mod n) e que, se n é primo, então alguma das potências

$$a^q,\ a^{2q},\ ...,\ a^{2^{k-1}q},\ a^{2^k q}$$

tem que ser congruente a -1 (mod n).

Quando não temos que $x \geqslant 1$, mas temos $x = 0$, então também temos que $a^q \equiv 1$ (mod n). Se nenhuma das coisas acontecem então concluímos que n é composto.

Mas se não sabemos que n é primo, o que acontece é que, se n não passa no teste, isto é, nem $a^q \equiv 1$ (mod n) e nem uma das potências da sequência é congruente a -1 (mod n),

então n certamente é composto. Todavia, se n passa no teste não é possível garantir que n é primo.

Mesmo sem essa garantia o teste de Miller-Rabin é mais preciso que o de Leibniz, uma vez que detecta os pseudoprimos para o testde de Leibniz como compostos. Por outro lado, o teste de Miller-Rabin também apresenta pseudoprimos, mas esses são chamados pseudoprimos fortes. O teste também é mais preciso em comparação ao de Leibniz porque, em geral, tem menos pseudoprimos fortes que a quantidade de pseudoprimos para o teste de Leibniz, considerando o mesmo intervalo. Por exemplo, para a base 2, e no intervalo de 1 a 10^9, o teste de Leibniz tem 5597 pseudoprimos, enquanto há apenas 1282 pseudoprimos fortes para o teste de Miller-Rabin nesse mesmo intervalo.

Além disso, o teste de Miller-Rabin tem a vantagem de que, se aplicado várias vezes, necessariamente para $(\frac{n}{4} + 1)$ vezes, com bases diferentes e passar no teste para todas elas, então n tem que ser primo. Isso também mostra que, se o número testado passar no teste uma única vez, tem a probabilidade de 75%, no mínimo, de ser primo.

Decerto que isso não é tão satisfatório assim, até porque 75% pode não ser uma probabilidade razoável. A situação também se complica mais se o número n é grande, porque aí é necessário testar cada vez mais para se ter a certeza de que n é primo.

3.5 TESTES DETERMINÍSTICOS

Até aqui, todos os testes apresentados não são seguros para verificar primalidade, mas são exatos para verificar se um número é composto. Vários outros testes têm a mesma finalidade e são bons em determinar se um número é composto.

Este é, então, um exemplo de tesde probabilístico. Garante certeza para números compostos e uma probabilidade de 75% para números primos. Existem outros testes probabilíticos não apresentados neste trabalho, como o teste de Solovay-Strassen, que usa conceitos que vão muito além dos objetivos propostos, mas que pode ser consultado em (FALEIROS, 2011).

Além dos testes probabilísticos, existem testes determinísticos, que como o nome já sugere, são testes capazes de determinar se um número é primo. Diferentemente dos testes probabilísticos, os testes determinísticos são capazes de responder com certeza se um número é primo ou composto.

O mais simples teste determinístico é o teste das divisões sucessivas, que consiste em dividir um dado número n por todos os números inteiros a tais que $2 \leqslant a < n$. Se n for divisível por pelo menos um número inteiro a que satisfaz $2 \leqslant a < n$, então n é composto. Se n não for divisível por nenhum deles, então n é primo .

Assim como o Crivo de Eratóstenes, o método das divisões sucessivas é extremamente demorado e pouco eficiente. Na verdade, não é necessário testar todos os números a tais que $2 \leqslant a < n$, mas é suficiente testar os números a tais que $2 \leqslant a \leqslant \sqrt{n}$. A proposição a seguir sustenta a afirmação.

Proposição 3.2. *Seja n um número natural composto, então n tem um divisor primo p tal que* $p \leqslant \sqrt{n}$.

Prova. Seja p o menor divisor primo de n, então $n = pa$ para algum $a \in \mathbb{N}$. Como $p \leqslant a$, por hipótese, temos que

$$p^2 \leqslant p \cdot a \Rightarrow p^2 \leqslant n \Rightarrow p \leqslant \sqrt{n}.$$

□

Garantindo que só é necessário testar os números a tais que $2 \leqslant a \leqslant \sqrt{n}$. Se nenhum deles dividir n, então n é primo.

A proposição poupa muito trabalho, mas não o suficiente para deixar o método eficiente, afinal ainda é necessário testar vários números, o que fica ainda pior se n for muito grande.

Outro teste determinístico é o teste de Lucas-Lehmer, que também extrapola os assuntos propostos neste trabalho, sendo necessário o conhecimento de teoria de grupos que está além dos objetivos, todavia o teste de Lucas-Lehmer, entre outros, pode ser encontrado em Coutinho (2014).

4 CRIPTOGRAFIA RSA

Depois de apresentados os testes de primalidade e os métodos inventados para tentar gerar primos, a entrada para a criptografia RSA está preparada. Neste capítulo serão apresentadas uma breve história do surgimento da criptografia RSA, o algoritmo RSA, o funcionamento e alguns exemplos. Mas antes é importante entender o que é a criptografia.

Em grego, “cryptos” significa oculto, secreto ou escondido. Logo a criptografia cuida dos métodos para tornar uma mensagem secreta, de forma tal que apenas seu destinatáio desejado seja capaz de interpretá-la corretamente.

4.1 A IDEIA DO RSA

No inícios dos anos 1980, dois matemáticos e um cientista da computação do Massachusetts Institute of Technology (MIT), respectivamente Ronald Linn Rivest (1947–), Adi Shamir (1952–) e Leonard Max Adleman (1945–) se interessaram no desenvolvimento de um sistema de criptografia de chave pública. A ideia inicial era de Rivest, que trouxe depois seus colegas para no fim desenvolverem o que ficou conhecido como sistema de criptografia RSA.

Muita história já havia se passado e já se conhecia a importância da troca de mensagens seguras, fosse de caráter pessoal, bélico, segurança nacional ou outros assuntos.

Antes da chegada dos primeiros computadores domésticos, o envio de mensagens secretas lidava essencialmente com um problema: tanto o emissor quanto o receptor da mensagem precisaraiam saber que método de codificação e decodificação da mensagem seria usado. Esse é o exemplo da máquina Enigma. Apesar da mensangem codificada ser transmitida por rádio e a distância não ser tão relevante, por causa da distância que as transmissões de rádio cobriam, era crucial que o receptor soubesse qual a posição dos rotores (a chave) para decodificar a mensagem recebida.

Bailey Whitfield Diffie (1944–) e Martin Hellman (1945–) publicaram um artigo que abriria novos caminhos para a criptografia e a segurança na internet: a criptografia de chave pública. Rivest e Shamir, deslumbrados pelas propostas de Diffie e Hellman, que eram a favor da ideia de que a criptografia e a segurança das mensagens deveriam beneficiar todas as pessoas e deveria ser algo público, não somente um privilégio dos detentores de grandes poderes ou entidades como o governo, resolveram estudar o assunto publicado no artigo (SAUTOY, 2007).

A ideia era possibilitar que não fosse mais necessário que o emissor e o receptor tivessem que usar a mesma chave, que fazia o papel da encriptação e desencriptação da mensagem. Para o emissor, conhecer a chave de encriptação seria suficiente e, por isso, poderia ser de conhecimento público, todavia não deveria ser possível que o código fosse quebrado, pelo menos não com tanta facilidade.

O artigo parecia algo muito teórico e difícil de ser aplicado no mundo real. Diffie e Hellman eram acadêmicos, mas isso não implicava que eles não entendessem as dificuldades

da implementação de um sistema de criptografia de chave pública na prática. Nem sempre é fácil colocar as coisas aprendidas na teoria em uma ação prática.

Mas Rivest via nessa situação uma porta nova se abrindo, exatamente com o que se interessava, a complexidade da teoria e a aplicação em situações reais.

Rivest e Shamir resolveram atacar a situação primeiro. Adleman ainda resistia à essa ideia, ele preferia as coisas mais abstratas, aqueles trabalhos e as tentativas de Rivest e Shamir de aplicar algo da teoria ao mundo real não era interessante para ele.

Com a chegada dos computadores, muitas tarefas que despenderiam muito tempo, como cálculos longos, puderam ser significativamente reduzidas de tempo. Muito provavelmente, se Turing dispusesse de um desses computadores, poderia realizar cálculos com menos esforço e de maneira muito mais rápida, possivelmente adiantando a quebra da cifra da Enigma.

A partir da insistência de Rivest e o interesse mais tarde por suas ideias, Adleman se juntou à dupla e formou-se o trio Rivest, Shamir e Adleman. Eles procuravam encontrar os problemas da matemática verdadeiramente difíceis e que pudessem se relacionar com as ideias de Diffie e Hellman, assim poderiam encontrar a solução para a implementação do sistema de criptografia de chave pública.

Problemas difíceis ou que levam muito tempo para se resolver era o que buscava o trio do MIT. Era na dificuldade de se resolver certos problemas da matemática que acreditavam que estaria a força da segurança do método de criptografia de chave pública que procuravam.

Constantemente as ideias que Rivest e Shamir tinham sobre o método para fazer o sistema de criptografia viável eram descartadas. Adleman era o responsável pelos testes e por refutar as ideias da dupla, afinal era essa sua especialidade e maior interesse, a teoria que seus colegas usavam para suas propostas.

Eventualmente Rivest mostrou a Adleman um manuscrito sobre sua nova proposta do sistema. Dessa vez decidiram consultar um especialista para ter certeza de que estavam mesmo num caminho promissor, pois o método que propuseram baseava-se na teoria dos números e na dificuldade da fatoração de números grandes.

> Rivest decidiu que eles deveriam descobrir se a fatoração era realmente tão difícil. “O problema da fatoração era uma forma de arte obscura naquela época. Havia pouca literatura sobre o assunto. Era difícil fazer boas estimativas sobre o tempo que os algoritmos já propostos levariam para resolvê-lo.” Um dos especialistas no assunto era Martin Gardner, um dos grandes responsáveis pela popularização mundial da matemática. Gardner ficou intrigado com a proposta de Rivest e perguntou se poderia publicar um artigo sobre aquela idéia em sua coluna regular na Scientific American (SAUTOY, 2007).

No fim das contas, estavam certos sobre suas ideias. Estavam preocupados que a fatoração dos números não fosse algo tão difícil assim, ainda mais com o uso dos computadores, mas acabou por ser algo realmente difícil de ser fazer naquela época e continua sendo até hoje.

Por fim, Rivest, Shamir e Adleman lançaram seu famoso sistema de criptografia de chave pública e o nomearam de RSA, o acrônimo composto das inicais dos sobrenomes de cada um

deles. A visão embrionária de Diffie e Hellman pode se tornar realidade nas mãos do trio do Massachusetts Institute of Technology.

A distância entre receptor e emissor já não seria um problema tão grande com a chegada internet. A questão central agora seria a segurança da informação transmitida pela internet. O RSA proporcionaria uma estabilidade à segurança de dados, pelo menos era essa a proposta.

4.2 O ALGORITMO RSA

Mas uma pergunta ainda não foi respondida. Afinal, como funciona o sistema RSA?

Para que as ideias de Diffie e Hellman se tornassem realidade e aplicáveis ao mundo real, os três colegas do MIT tiveram muito trabalho em encontrar os problemas matemáticos de difícil resolução para que também o sistema de criptografia fosse difícil de ser quebrado. É nesse sentido que a teoria dos números, nos séculos passados não vista com tanta aplicabilidade para o mundo real, pode fazer uma das suass maiores contribuições para a criptografia e ganhar os holofotes.

O RSA se apoia em uma questão matemática muito antiga, a fatoração dos números. Até hoje não se conhece um algoritmo eficientemente rápido para a fatoração de números.

Os capítulos anteriores já mostraram os grandes esforços de vários matemáticos brilhantes para encontrar a ordenação dos primos, um algoritmo que pudesse gerar primos ou mesmo algoritmos que pudessem verificar a primalidade de um determinado número. Apesar de alguns conseguirem descobrir um algoritmo que detectasse a primalidade de números ou que acusasse que ele é composto, mesmo que a um custo de tempo exageradamente longo para números muito grandes, nenhum deles conseguiu um algoritmo eficiente para a fatoração de inteiros.

Como um número primo é aquele que só é divisível por 1 e por si mesmo (se tomarmos apenas os positivos), então jamais se encontraria um fator de um número primo grande maior que 1 senão o próprio número. Se esse número for suficientemente grande, então mesmo as máquinas de calcular ou computadores muito rápidos podem demorar muito tempo para poder determinar se um número é primo ou não.

Quando enviamos uma mensagem pela internet, para que ela seja criptografada pelo RSA, o computador usa essa mensagem e faz um determinado cálculo, relativamente de fácil realização, afinal estamos falando de uma máquina. Esse cálculo é que criptografa a mensagem e deixa ela ilegível (até mesmo para o emissor). A partir daí, somente o receptor deve ser capaz de descriptografar a mensagem e poder lê-la porque, apesar de o cálculo para critografar a mensagem ser fácil, o processo inverso é quase que impossível de se fazer sem a chave que tem esse fim, mesmo para a máquina que acabou de criptografar.

Isso pode parecer deveras estranho, mas é algo relativamente muito simples e a matemática por trás disso pode explicar esses motivos.

É exatamente no fato de ser extremamente difícil de descriptografar a mensagem RSA sem a chave específica para esse fim que se apoia a maior força desse sistema, e deve ser

realmente forte, porquanto tanto a chave para a criptação é pública quanto a mensagem pode se tornar pública ou ser alvo de ataques, se for inteceptada por um outro usuário da rede.

A base matemática necessária para a aplicação do RSA é composta de alguns elementos já vistos anteriormente: a divisibilidade de números, as propriedades dos números primos, as relações de congruência módulo m, o Teorema de Euler, que é a generalização do Pequeno Teorema de Fermat e a função φ de Euler. A segurança do RSA está apoiada principalmente na dificuldade da fatoração de inteiros.

Para codificar usando o RSA precisamos de um número n que será o produto de dois números primos e de um outro número positivo e que tenha a propriedade de ser inversível módulo $\varphi(n)$, isto é mdc$(e, \varphi(n)) = 1$.

Já foi mostrado que calcular mdc pode ser uma tarefa difícil, mas quando se conhecem os números p e q, que são os fatores de n, isso é muito simples, já que $\varphi(n) = (n-1)$ quando n é primo e $\varphi(n) = (p-1)(q-1)$ quando se conhecem os primos p e q.

Para a aplicação do algoritmo é necessário escolher dois números primos distintos e grandes p e q, isto é, p e q têm vários algarismos. A seguir calcula-se o produto desses dois primos $pq = n$.

Depois, deve-se escolher um número natural e de forma que mdc $(e, (p-1)(q-1)) = 1$.

E um número inteiro d de forma que d satisfaça duas condições:

a) $ed \equiv 1 \ (mod \ (p-1)(q-1))$.

b) $1 \leq d < (p-1)(q-1)$.

A chave para o funcionamento do sistema RSA depende dos números n, e e d. O par (n, e) constitui a chave pública, que deve ser de conhecimento de qualquer um que se deseje ser emissor da mensagem. O par (n, d) é a chave privada e secreta. O receptor deve manter em segredo tanto o número d quanto os números p e q que geraram n.

De posse da chave púbica, a máquina do usuário faz os cálculos para encriptação e envia a mensagem para o receptor, este último usa sua chave para poder ler a mensagem original.

4.3 CODIFICAÇÃO E DECODIFICAÇÃO

Para se ter uma ideia melhor do funcionamento do algoritmo, esta seção apresenta um exemplo prático do RSA. Apesar de ser recomendado escolher números primos grandes, serão escolhidos números primos pequenos apenas para facilitar os cálculos, sem prejuízo para o entendimento do sistema na ocasião do uso de números primos grandes.

Para termos um exemplo mais próximo do mundo real escolheremos um texto escrito com as letras do nosso alfabeto para sofrer o processo descrito na seção anterior.

Tomemos o texto: “DEUS É FIEL”.

Como faremos cálculos com números, convém transformarmos o texto do nosso alfabeto em números, assim vamos trabalhar com elementos da mesma classe.

Usaremos a correspondência abaixo do alfabeto para números.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
10	11	12	13	14	15	16	17	18	19	20	21	22	23

O	P	Q	R	S	T	U	V	W	X	Y	Z
24	25	26	27	28	29	30	31	32	33	34	35

Assim usamos todas as letras do nosso alfabeto. Para poder separar uma palavra da outra, na nossa língua, usamos um espaço, esse espaço será denotado pelo número 99.

Vale observar que a correspondência não foi iniciada pelo número 0 ou pelo número 1 nem por nenhum outro número que contivesse apenas um algarismo. Não é simplesmente um capricho, mas tem sua razão: evitar ambiguidades.

Se usássemos, por exemplo, a correspondência

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	2	3	4	5	6	7	8	9	10	11	12	13	14

não seria possível distinguir se 13 correspondria a AC ou M.

Agora façamos a correspondência das letras do texto em números.

D	E	U	S		E		F	I	E	L
13	14	30	28	99	14	99	15	18	14	21

E nossa frase convertida será 13143028991499151814 21.

Agora que já convertemos a frase em números fica mais fácil trabalhar com o RSA, mas ainda falta a essência desse sistema: escolher os números que vamos usar como chave.

Como já mencionado antes, vamos escolher números primos pequenos para facilitar os cálculos. Escolhamos $p = 11$ e $q = 19$, dessa maneira temos que $n = 209$.

Agora podemos quebrar o bloco numérico da mensagem em pequenas partes, tomando alguns cuidados:

a) Por segurança, é bom garantir que os blocos não correspondam apenas a unidades linguísticas, isto é, a letras do alfabeto utilizado, para que isso não facilite a decifragem por uma contagem de frequência.

b) Os blocos não devem começar pelo algarismo 0. Problemas podem surgir na hora de fazer os cálculos pois, por exemplo, somar 037 a um determinado número é igual a somar com 37, mas para o código linguístico, 037 pode não ser a mesma coisa que 37, principalmente na conversão dos blocos para a unidade linguística e na operação de congruência.

c) Os números dos blocos não devem ser maiores que n.

Tomando esses cuidados podemos dividir nossa mensagem em blocos da seguinte maneira:

$$131 - 4 - 30 - 28 - 99 - 149 - 9 - 151 - 8 - 14 - 21.$$

Note que o maior número dos blocos divididos é 131 que é menor que 209 = n. Outra observação importante é que a maneira de se divir em blocos não é única, basta que se tome os devidos cuidados.

Passando para a parte da codificação, vamos precisar de (n, e) que é a chave pública. Pelas regras, devemos escolher e tal que seja inversível módulo $\varphi(n)$, que quer dizer mdc $(e, \varphi(n)) = 1$.

Porque escolhemos p e q, é fácil calcular $\varphi(n)$.

$$\begin{aligned}\varphi(n) = \varphi(209) &= \varphi(11 \cdot 19)\\ &= \varphi(11)\varphi(19)\\ &= (11-1)\cdot(19-1)\\ &= 10 \cdot 18 = 180\end{aligned}$$

Para escolher e no nosso exemplo não é tão difícil, basta escolher o menor primo que não divide 180, portanto $e = 7$ satisfaz a condição mdc $(e, \varphi(n)) = 1$.

Assim, temos a chave de codificação pública e basta codificar cada bloco que foi dividido com a chave obtida. Vale lembrar que não se deve juntar depois os blocos codificados a fim de formar um número grande, como fizemos no processo de transformar a frase em um número. Se isso acontecer, não seremos capazes de decodificar a mensagem.

Chamando de b um dos blocos ainda não codificados e de $C(b)$ esse bloco já codificado, o que nos falta é calcular $C(b)$, que é o resto da divisão de b^e por n.

Agora resta codificar cada bloco separadamente. Tomando o primeiro bloco da mensagem, o bloco 131, temos

$$131^7 \equiv 43 \pmod{209}.$$

Logo, $C(131) = 43$. Codificando toda a mensagem, temos

$$43 - 82 - 68 - 118 - 44 - 74 - 4 - 189 - 46 - 174 - 109.$$

Agora com os blocos codificados, já é possível enviar a mensagem ao destinatário, que deverá fazer o precosso de decodificação usando o método mostrado a seguir.

Chamando de a um dos blocos codificados, $D(a)$ será o bloco decodificado com a chave privada (n, d).

Para encontrar d nas condições estipuladas é suficiente que $ed \equiv 1 \pmod{180}$ e $1 \leq d < 180$, que é o equivalente a calcular o inverso de e em $\varphi(n)$. Reescrevendo a congruência de uma forma conveniente e substituindo o número e já determinado, teremos

$$7d = 1 + 180k.$$

para algum k inteiro. Devemos econtrar a solução da equação e para isso vamos reescrever a equação de uma maineira mais analítica.

$$d = \frac{1 + 180k}{7} = \frac{1}{7} + \frac{5k}{7} + 25k = \frac{1 + 5k}{7} + 25k.$$

Como precisamos de d inteiro, a expressão do lado direito da igualdade acima deve ser um número inteiro. Existem infinitas possibilidades para k de forma que d seja inteiro, logo precisamos de uma restrição, que é a condição $1 \leq d < 180$.

Verificamos que $d = 180n + 103$ e $k = 7n + 4$ com n um inteiro satisfaz a equação, todavia, tomando apenas os n inteiros não negativos, o único valor de n possível que satisfaz a equação e as restrições dadas é $n = 0$, pois $k = 7(0) + 4 = 4$ e

$$\frac{1 + 5 \cdot (4)}{7} + 25 \cdot (4) = \frac{1 + 20}{7} + 100 = 3 + 100 = 103.$$

É fácil ver que para $n = 1$ a condição não é satisfeita, pois se $n = 1$, então $d = 283 > 180$. Logo, $d = 103$.

Para decodificar, por exemplo, o bloco 118 da mensagem codificada, basta calcular a forma reduzida de 118^{103} (mod 209), que é o resto da divisão de 118^{103} por 209.

$$118^{103} \equiv 28 \ (mod \ \ 209).$$

Logo, $D(118) = 28$. Assim a mensagem decodificada fica:

$$131 - 4 - 30 - 28 - 99 - 149 - 9 - 151 - 8 - 14 - 21.$$

Idêntica à mensagem inicial antes da encriptação.

Agora resta apenas juntar os blocos formando um número grande novamente e depois dividir em blocos de 2 algarismos, uma vez que essa é a correspondência do alfabeto usada.

O algoritmo RSA também permite que o possuidor da chave privada envie mensagens criptografadas para os possuidores da chave pública com o intuito de que apenas os esses usuários, de posse da chave pública, consigam ler a mensagem. A diferença é que a mensagem não fica tão secreta assim.

Isso é possível principalmente pela escolha e restrições dos números n, e e d. A condição de e ser inversível em $\varphi(n)$ e d ser este inverso é que garante que seja possível fazer o processo inverso.

4.4 A GARANTIA DO MÉTODO

Depois de entender como o método de criptografia RSA funciona pode surgir uma questão importante - Será que ele funciona sempre? A resposta é sim, se os cuidados descritos anteriormente forem tomados.

Verifiquemos que $DC(b) = b$.

Primeiro recordemos os cuidados que foram tomados.

a) $n = pq$, com p e q primos.

b) e é tal que mdc $(e, \varphi(n)) = 1$, invesível módulo n.

c) d é o inverso de e (mod $\varphi(n)$).

d) b é um número inteiro tal que $1 \leq b \leq n - 1$.

Pela definição de $DC(b)$, temos que

$$DC(b) \equiv (b^e)^d \equiv b^{ed} \pmod{n}.$$

Do item 3 acima segue que

$$ed = 1 + k\varphi(n) = 1 + k(p-1)(q-1),$$

para algum k inteiro maior que 0. Necessariamente $k > 0$ porque $e > 2$ e $d > 2$. $\varphi(n) > 0$ e par. Se k fosse um inteiro menor ou igual a 0, então $2 < ed = 1 + k\varphi(n) \leq 1$, o que é uma contradição.

Substituindo $ed = 1 + k\varphi(n)$ em $b^{ed} \pmod{n}$, temos

$$b^{ed} \equiv b^{1+k\varphi(n)} \equiv b(b^{k\varphi(n)}) = b(b^{\varphi(n)})^k \pmod{n}. \qquad (4.1)$$

Pelo o Teorema de Euler $b^{\varphi(n)} \equiv 1 \pmod{n}$, se mdc $(b, n) = 1$.

Para verificar que mdc $(b, n) = 1$ calcularemos a forma reduzida de b^{ed} (mod p) e b^{ed} (mod q).

De (4.1), temos que

$$b^{ed} \equiv b(b^{p-1})^{k(q-1)} \pmod{p}.$$

Sabemos, das condições impostas que $b < n = pq$, mas isso não garante que $p \nmid b$.

Se $p \nmid b$, então, pelo Teorema de Fermat, $b^{p-1} \equiv 1$ (mod p), donde $b^{ed} \equiv b$ (mod p).

Se $p \mid b$, então $b \equiv 0$ (mod p,) e $b^{ed} \equiv b$ (mod p). Logo,

$$b^{ed} \equiv 1^k b \pmod{n} \equiv b \pmod{n}.$$

Portanto

$$DC(b) \equiv b \ (mod\ n).$$

A demosntração para q é análoga e será omitida.

É verdade que mostramos que $DC(b) \equiv b \ (mod\ n)$, mas não mostramos que $DC(b) = b$.

Com todos os resultados que tivemos, não fica difícil mostrar que $DC(b) = b$. De fato, como $DC(b)$ e b são todos números estritamente menores que n, segundo os critérios de ecolhas e como acabamos de mostrar que $DC(b) \equiv b \ (mod\ n)$, só podem ser congruentes módulo n se $DC(b) = b$, pois só são congruentes se deixam mesmo resto.

Com isso mostramos que o algoritmo sempre funciona.

5 A SEGURANÇA DO RSA

Neste capítulo são mostrados os pontos fortes e fracos da segurança do RSA.

5.1 A CHAVE PRIVADA

A segurança do RSA baseia-se principalmente no segredo da chave privada, uma vez que a outra chave é pública e de acesso fácil para os outros usuários do sistema.

O par (n, e), nas condições do capítulo anterior, é denotado chave pública, o par (n, d) é a chave privada. A segurança reside em ser difícil calcular d apenas conhecendo n e e, pois o que acontece na prática é que só conseguimos calcular o número d a partir do algoritmo estendido euclidiano aplicado a $\varphi(n)$ e e.

De fato calcular $\varphi(n)$ de um número primo é simples se n for conhecido, mas os capítulos anteriores mostraram que não existe um algoritmo eficiente para calcular $\varphi(n)$ se n não é primo, e é exatamente esse o caso, porque $n = pq$.

Para calcular $\varphi(n)$ de n composto é necessário fatorar n, e aí está mais um ponto forte do RSA: não se conhece um algoritmo rápido para fatorar numeros grandes. Vale lembrar que as escolhas de p e q devem ser números grandes justamente para que n seja um número ainda maior e dificulte a fatoração. Portanto, na prática, o código deve ser quebrado apenas se conseguirmos fatorar n.

Como até hoje não se conhecem tais algoritmos para fatoração rápida, acredita-se que descobrir tal algoritmo ou quebrar o sistema RSA é equivalente.

Ainda seria possível pensar que conseguiríamos encontrar b que é o bloco da mensagem não codificada, nas condições do capítulo anterior, a partir da forma reduzida de $b^e \pmod{n}$, pois tanto n, e e a forma reduzida são conhecidos. O problema é que isso é impraticável por tentativas quando n é muito grande e até hoje não existe um método eficiente (COUTINHO, 2014).

5.2 A ESCOLHA DOS NÚMEROS PRIMOS

Até aqui sabemos onde reside a maior força do RSA, a dificuldade em fatorar n, mas existe outro fator bastante importante, a escolha de p e q, pois são eles quem determinam n.

Não basta escolher apenas p e q grandes, há um cuidado bastante importante que se deve tomar na escolha dos dois números prmos. Eles não devem ser tão próximos, isto é, $|p - q|$ não deve ser muito pequeno.

A escolha de p e q tais que $|p - q|$ seja pequeno torna mais fácil o trabalho da fatoração de n. Usando o algoritmo para fatoração de Fermat, que é bastante eficiente quando n tem um fator que não é muito menor que $\sqrt{n}$, que é justamente o que acontece quando $|p - q|$ é pequeno, pois quanto mais próximo forem p e q mais próximo fica p^2 ou q^2 de n, o que implica

que mais próximos ficam p e q de $\sqrt{n}$.

A ideia do algoritmo de fatoração de Fermat é encontar números positivos j e k tais que $n = j^2 - k^2$, pois se encontrados esses números, então encontraremos os fatores de n, sendo um fator $(j - k)$ e outro fator $(j + k)$. De fato

$$n = j^2 - k^2 = (j - k)(j + k).$$

Supondo que seja fácil encontrar a raiz quadrada de um dado número, ou pelo menos a parte inteira dela, que é o que o algoritmo de Fermat usa, pode-se encontrar rapidamente os fatores desse número, claro, usando computadores.

Supondo que n é composto e denotando a parte inteira da raiz de n por $[\sqrt{n}]$, por exemplo $[\sqrt{16}] = 4$, $[\sqrt{9}] = 3$ e $[\sqrt{14}] = 3$, começa-se com $j = [\sqrt{n}]$. Se n é um quadrado perfeito, então $j = [\sqrt{n}] = \sqrt{n}$. Nessas condiçõeso $n = j^2$ e j é fator de n, que é o caso mais simples.

Vale notar que nesse caso temos $k = j$ e $|j - k| = 0$.

Se $j \neq [\sqrt{n}]$, então somamos uma unidade ao número j e calculamos o número k tal que $k = \sqrt{(j + 1)^2 - n}$. Se k for um inteiro que satisfaz a igualdade, então os fatores de n foram encontrados e são $((j + 1) - k)$ e $((j + 1) + k)$.

Se k não for um inteiro, então acrescentamos mais uma unidade ao número j e calculamos $k = \sqrt{(j + 1 + 1)^2 - n}$, assim sucessivamente até encontrar k inteiro que satisfaça a igualdade ou encontrar $j = \dfrac{n+1}{2}$.

Se encontrarmos k inteiro que satisfaz a iguadade com $j \underbrace{+1 + ... + 1}_{m \text{ vezes}} < \dfrac{n+1}{2}$, então encontramos os fatores de n.

De fato, pois, se encontrarmos k inteiro satisfazendo

$$k = \sqrt{(j \underbrace{+1 + ... + 1}_{m \text{ vezes}})^2 - n},$$

sendo m o número de vezes que foi somada uma unidade ao número j, então basta tomar $j = j_m$, sendo $j_m = j \underbrace{+1 + ... + 1}_{m \text{ vezes}}$. Nessas condições, é claro que j tem que ser maior que $\sqrt{n}$, mas não é imediato que j deve ser menor que $\dfrac{n+1}{2}$.

De fato, pois, supondo $n = xy$ composto e $x \leqslant y$ desejamos encontrar j e k tais que

$$n = xy = (j - k)(j + k) = j^2 - k^2 \tag{5.1}$$

Como, por hipótese, $j - k \leqslant j + k$, pode-se tomar $x = j - k$ e $y = j + k$, donde

$$j = \frac{x + y}{2} \quad , \quad k = \frac{y - x}{2} \tag{5.2}$$

substituindo em (5.1), temos

$$n = xy = (j-k)(j+k) = j^2 - k^2 = \left(\frac{x+y}{2}\right)^2 - \left(\frac{y-x}{2}\right)^2 \tag{5.3}$$

Vale notar que, por hipótese n é ímpar, pois é produto de dois primos grandes, logo diferentes de 2 e também ímpares. Então x e y são ímpares e $(x+y)$ e $(x-y)$ são pares, consequentemente $\frac{x+y}{2}$ e $\frac{y-x}{2}$ são inteiros.

Assim, o algoritmo de fatoração de Fermat dá uma resposta para os fatores de n testando os números entre $\sqrt{n}$ e $\frac{n+1}{2}$, isto é

$$[\sqrt{n}] \leqslant \frac{x+y}{2} \leqslant \frac{n+1}{2}$$

Para demonstrar esse fato, inicialmente usaremos as desigualdades

$$[\sqrt{n}] \leqslant \frac{x+y}{2} < \frac{n+1}{2}.$$

Mostraremos posteriormente que, se $\frac{x+y}{2} = \frac{n+1}{2}$, então n tem que ser primo.

Proposição 5.1. *Seja $n = xy$ um número composto com $x \leqslant y$ inteiros positivos. Então*

$$[\sqrt{n}] \leqslant \frac{x+y}{2} < \frac{n+1}{2}.$$

Prova. Se $x = y$ já vimos que n é um quadrado perfeito e já encontramos um fator $x = \sqrt{n} = [\sqrt{n}]$.

Se $x \neq y$ e, sem perda de generalidade, $x < y$, podemos escrever $1 < x < y < n$.

Da desigualdade $\frac{x+y}{2} < \frac{n+1}{2}$, obtemos que $(x+y) < n+1$.

Da hipótese $n = xy$, podemos substituir n na expressão $(x+y) < n+1$ obtendo $x + y < (xy) + 1$.

Subtraindo $y + 1$ em cada lado da desigualdade, obtemos

$$x + y - (y+1) < (xy) + 1 - (y+1) \Rightarrow x + y - y - 1 < xy + 1 - y - 1 \Rightarrow x - 1 < xy - y.$$

.

Reescrevendo o produto da desigualdade, obtemos $x - 1 < (x-1)y$.

Como $x > 1$, por hipótese, podemos dividir os membros da igualdade por $(x-1)$ e obtemos $1 < y$.

Logo, $\frac{x+y}{2} < \frac{n+1}{2}$ é equivalente a $1 < y$.

Por hipótese, $1 < x < y < n$ e concluímos que $\frac{x+y}{2} < \frac{n+1}{2}$ é verdadeiro.

Da desigualdade $[\sqrt{n}] \leqslant \frac{x+y}{2}$ basta verificar que $\sqrt{n} \leqslant \frac{x+y}{2}$, pois $[\sqrt{n}] \leqslant \sqrt{n}$, valendo a igualdade se, e somente se, n é um quadrado perfeito, caso que já foi discutido.

$\sqrt{n} \leqslant \frac{x+y}{2}$ é válida se, e somente se, $n \leqslant \frac{(x+y)^2}{4}$.

Por (5.3), segue que

$$\left(\frac{x+y}{2}\right)^2 - \left(\frac{y-x}{2}\right)^2 = n \Rightarrow \left(\frac{x+y}{2}\right)^2 - n = \left(\frac{y-x}{2}\right)^2$$

e $\left(\frac{y-x}{2}\right)^2$ é sempre um número não negativo.

Concluímos que $\left(\frac{x+y}{2}\right)^2 - n \geqslant 0$, que é equivalente a $[\sqrt{n}] \leqslant \frac{x+y}{2}$.

□

Isso nos garante que, nos moldes de (5.2), se $n = xy$, sempre encontramos $j = \frac{x+y}{2}$ antes de chegar a $j = \frac{n+1}{2}$.

E se $j = \frac{x+y}{2}$, então

$$k^2 = \left(\frac{x+y}{2}\right)^2 - n = \left(\frac{y-x}{2}\right)^2.$$

Se encontrarmos $j = \frac{n+1}{2}$, então n é primo e encontraríamos, por exemplo $x = 1$ e $y = n$, daí $j = \frac{n+1}{2}$, pois supomos $j > 1$.

Computacionalmente, esse algorítmo não demanda tanto tempo ou processamento quanto o algoritmo para encontrar a fatoração de um número grande e por isso precisamos escolher de maneira inteligente os primos p e q.

O tamanho mínimo recomendado para o número n a fim de se construir uma chave pessoal é de 768 bits, que significa aproximadamente 231 algarismos. Para conseguir n desse tamanho, p e q devem ter mais que 100 algarismo ambos, para que sejam suficientemente grandes cada um deles, aproximadamente 104 e 127 algarismos respectivamente (COUTINHO, 2014).

Para escolher primos que tornem o algoritmo da fatoração de Fermat impraticável é importante que se escolham p e q tais que $|p - q|$ não seja muito pequeno, então uma boa estratégia é escolher p entre $\frac{4r}{10}$ e $\frac{45r}{100}$ e q próximo a $\frac{10^r}{p}$, sendo r o número de dígitos desejados aproximadamente para n.

Ainda assim o trabalho de se escolher os números primos não é fácil, precisamos conhecer ou descobrir primos com muitos algarismos porque, geralmente, quanto maiores forem esse números primos mais seguro será o sistema.

Além da escolha há outro problema que deve ser levado em consideração, os números $p - 1$, $p + 1$, $q - 1$ e $q + 1$ não devem ter fatores primos muito pequenos, pois facilitaria a descoberta de p e q por meio de outros algoritmos como o método $p - 1$ de Pollard que pode ser consultado em (ROSEN, 2014).

5.3 ASSINATURA

Os elementos basilares da segurança do sistema RSA foram apresentados segundo o olhar matemático, mas na prática ainda se deve tomar outros cuidados na implementação e uso. Um exemplo pode ilustrar bem uma situação em que apenas as precauções do ponto de vista matemático podem não ser suficientes.

Com a chave pública de fácil acesso, Renato recebeu uma mensagem pelo sistema RSA com os dizeres “Olá, Renato. Meu nome é Cássia e preciso te encontrar no saguão do hotel hoje, às dez horas pontualmente, para entregar-te uma maleta com sete mil reais”. Como Renato pode ter certeza de que a mensagem é realmente de Cássia? Para responder essa questão existe a assinatura da mensagem.

Já foi mostrado anteriormente que com as chaves públicas e privadas, tanto o emissor quanto o destinatário podem enviar e receber mensagens criptografadas. Antes não fazia tanto sentido o possuidor da chave privada enviar mensagens, mas com a assinatura isso muda.

Na seção 4.3 foi mostrado que, pelas escolhas dos números p, q, e e d, é possível aplicar o processo inverso de criptografia em relação às chaves, isto é, o possuidor da chave privada criptografa uma mensagem e envia para os possuidores das chaves públicas, que são capazes de ler a mensagem. Nesse fato também se baseia a assinatura.

Para que o destinatátio da mensagem consiga verificar se a mensagem que recebeu é mesmo do remetente que esperava, basta que o remetente insira uma assinatura na mensagem. Para que consiga fazer isso é necessário que também possua um par de chaves.

Nos moldes do capítulo 4, vamos chamar de C_r a função de codificação de Renato e D_r a função de decodificação de Renato. Assim, chamando de a o bloco da mensagem a ser codificada, quem quiser enviar uma mensagem para Renato deve codificar com $C_r(a)$ e enviá-la.

No exemplo apresentado, para que Renato seja capaz de verificar que a mensagem é de Cássia, ela deve incluir algo na mensagem que seja capaz de certificar sua autoria: a assinatura.

Cássia, possuindo um par de chaves, sendo C_c a função de codificação de Cássia e D_c a função de decodificação de Cássia, deve primeiro aplicar a sua função de decodificação à mensagem e depois a função de codificação de Renato, necessariamente nessa ordem, e somente depois enviar. Quando Renato receber a mensagem ele deve aplicar a sua função de decodificação primeiro e assim obtém $D_c(a)$ e a esse bloco aplica a função de codificação de Cássia, se a mensagem fizer sentido então ele pode assumir que é mesmo de Cássia, caso contrário não deve confiar na mensagem.

Segundo Coutinho (2014), as chances de uma mensagem enviada com assinatura fazer sentido sendo de outra pessoa com outro par de chaves é praticamente zero, principalmente se ela for uma mensagem longa, porque os elementos linguíticos precisam fazer sentido no texto todo.

EPÍLOGO E CONSIDERAÇÕES FINAIS

A criptografia RSA é razoavelmente recente, com suas ideias iniciais começando nos anos 1970, mas que usam resultados muito antigos, desde antes de Cristo, como a demonstração de infinitos primos, por Euclides, afinal se não existissem infinitos primos esse sistema estaria condenado.

Este trabalho abordou aspectos da criptografia RSA relacionados à matemática, mas há outros aspectos a serem considerados. Existem vários outros algoritmos não apresentados aqui para teste de primalidade, sejam eles deterministicos ou probabilísticos, ou algoritmos para encontrar fatores primos de números grandes.

Além disso, o RSA pode sofrer vários tipos de ataques, por exemplo, se o expoente d ou e das chaves sejam pequenos existe um algoritmo eficiente para calcular d a partir de n e e, ou um ataque temporal, baseado no tempo que o sistema leva para decodificar uma mensagem. O ataque por força bruta, que é simplesmente testar todas as combinações de chaves possíveis para decifrar uma mensagem, completamente inviável, mas ainda possível. Quase todos esses ataques, excetuando-se o de força bruta, exigem muito mais detalhes matemáticos e computacionais e por isso não foram abordados.

As principais desvantagens de se usar o RSA é a lentidão do método e a escolhas dos números que geram as chaves, todavia a segurança e compartilhamento de informações é razoavelmente mais segura que algumas criptografias simétricas.

Por exemplo, tentar fatorar um número de mais de 300 dígitos poderia requerer mais que 1 milhão de anos e mesmo que os computadores se tornem muito mais velozes no processamento desses cálculos, muito tempo terá passado, tempo suficiente para que se escolha um outro par de chaves.

O assunto da eficiencia da máquina de processamento, tanto para o processo de criptografia quanto para o processo de quebra do código não foi abordado por exigir mais informação computacional e tecnológica, que também extrapolam o proposto para o trabalho

Mesmo a criptografia não pode proteger todos os aspectos da informação. Por exemplo, se um intruso do sistema criar novos arquivos ou apagá-los antes que se aplique a criptografia ou modifique o programa para mudar a chave, nada se pode fazer.

Há ainda métodos e técnicas que se misturam ao RSA para deixá-lo cada vez mais seguro, como aplicações de funções *hash* que dificultam ainda mais a recuperação dos números que gerarm as chaves

Por se basear em problemas matemáticos para os quais não são conhecidos algoritmos eficientes de resolução, como a fatoração de números grandes, o RSA ainda é considerado seguro, todavia o sistema poderia se tornar completamente inútil, em questão de segurança, se algum dia alguém conseguir descobrir um método rápido para a fatoração de números grandes.

REFERÊNCIAS

CASTRO, J. K. S. **Teoria dos números**. Fortaleza: UAB/IFCE, 2010. Citado na página 12.

COUTINHO, S. C. **Números Inteiros e Criptografia RSA**. 2. ed. Rio de Janeiro: IMPA, 2014. Citado 5 vezes nas páginas 24, 26, 36, 39 e 40.

EUCLIDES. **Os Elementos**. São Paulo: Editora da UNESP, 2009. Citado na página 10.

FALEIROS, A. C. **Criptografia**. São Carlos: SBMAC, 2011. Citado na página 25.

FEBRABAN. **Pesquisa FEBRABAN de Tecnologia Bancária**. 2020. Citado na página 8.

ROSEN, K. H. **Elementary Number Theory and Its Applications**. 6. ed. Massachusets: Addison-Wesley, 2014. Citado 3 vezes nas páginas 10, 11 e 39.

SAUTOY, M. du. **A música dos números primos**. Rio de Janeiro: Jorge Zahar, 2007. Citado 6 vezes nas páginas 11, 17, 19, 21, 27 e 28.

SHOKRANIAN, S. **Criptografia para iniciantes**. Brasília: Editora da UNB, 2005. Citado na página 19.

SINGH, S. **The Code Book: the secret history of codes and code-breaking**. London: Fourth Estate, 1999. Citado na página 19.