UNIVERSIDADE ESTADUAL DE PONTA GROSSA
SETOR DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA
MESTRADO PROFISSIONAL - PROFMAT

JOSÉ ROBYSON AGGIO MOLINARI

NÚMEROS PRIMOS E A CRIPTOGRAFIA RSA

PONTA GROSSA
2016

JOSÉ ROBYSON AGGIO MOLINARI

NÚMEROS PRIMOS E A CRIPTOGRAFIA RSA

Dissertação apresentada ao Programa de Pós-Graduação em Matemática PROFMAT - UEPG como parte dos requisitos para obtenção do título de Mestre em Matemática.

Orientadora: Prof. Dra. Fabiane de Oliveira

PONTA GROSSA

2016

Ficha Catalográfica
Elaborada pelo Setor de Tratamento da Informação BICEN/UEPG

Molinari, José Robyson Aggio
M722 Números primos e a criptografia RSA/ José Robyson Aggio Molinari. Ponta Grossa, 2016.
54f.

Dissertação (Mestrado Profissional em Matemática em Rede Nacional - Área de Concentração: Matemática), Universidade Estadual de Ponta Grossa.
Orientadora: Profª Drª Fabiane de Oliveira.

1.Criptografia RSA. 2.Números Primos. 3.Função de Euler. I.Oliveira, Fabiane de. II. Universidade Estadual de Ponta Grossa. Mestrado Profissional em Matemática em Rede Nacional. III. T.

CDD: 512.723

TERMO DE APROVAÇÃO

José Robyson Aggio Molinari

"NÚMEROS PRIMOS E A CRIPTOGRAFIA RSA"

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Mestrado Profissional em Matemática em Rede Nacional da Universidade Estadual de Ponta Grossa, pela seguinte banca examinadora.

Orientador: Profa. Dra. Fabiane de Oliveira
Departamento de Matemática, UEPG/PR

Profa. Dr. Márcio André Martins
Departamento de Matemática, UNICENTRO/PR

Prof. Dr. Giuliano Gadioli La Guardia
Departamento de Matemática, UEPG/PR

Ponta Grossa, 03 de Fevereiro de 2016.

AGRADECIMENTOS

À Deus pela oportunidade concedida.

À minha mãe Ilda Aggio, pela educação e incentivo aos estudos.

À minha noiva Franciéle Retslaff pelo amor, paciência e compreensão.

Aos professores do PROFMAT por compartilhar seus conhecimentos.

À minha orientadora Fabiane de Oliveira pelo apoio e orientação neste trabalho.

Aos amigos do Mestrado pela troca de conhecimentos adquiridos.

“Os encantos dessa sublime ciência se revelam
apenas àqueles que tem coragem de irem a fundo nela”

Carl Friedrich Gauss

RESUMO

Este trabalho apresenta alguns métodos de criptografia utilizados na antiguidade e também o avanço na maneira de criptografar. O objetivo principal é o estudo do Método RSA: contextualização histórica, a importância dos números primos, a ineficiência dos algoritmos de fatoração, codificação, decodificação, a segurança e um estudo sobre a função de Euler. Desenvolveu-se algumas atividades com conteúdos matemáticos relacionadas à criptografia. Desta maneira, espera-se que esta pesquisa possa apresentar uma metodologia auxiliar para o ensino de certos conteúdos da matemática, articulados com a utilização da criptografia.

Palavras-chave: Criptografia RSA, Números Primos, Função de Euler.

ABSTRACT

This study presents some of the encryption methods used in antiquity as well as the advance in the way of encrypting. The main objective of this work is the study of RSA Method: its Historical context, the importance of prime numbers, the inefficiency of factorization algorithms, coding, decoding, its security and a study of the Euler function. Some activities with mathematical content related to encryption have been developed. Thus, it is expected that this research can present an auxiliary methodology for teaching certain math content, linked to the utilization of cryptography.

Keywords: RSA encryption, Prime Numbers, Euler function.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

INTRODUÇÃO

Com o avanço dos meios de comunicação e tecnológicos, tornou-se necessário o desenvolvimento de métodos seguros de transmissão de informações, ou seja, métodos de codificação de mensagens. Com isso surgiu a criptografia de chave pública, também conhecida por criptografia assimétrica. Esse método possui duas chaves distintas que são utilizadas. Uma delas a chave pública, que está disponível para qualquer pessoa, ou seja, é de conhecimento de todos e é utilizada para codificar as mensagens, que só poderão ser decodificadas por quem possuir a chave privada correspondente.

O mais conhecido dos métodos de criptografia de chave pública é o RSA. Este método foi inventado em 1977 por R.L. Rivest, A. Shamir e L. Adleman, que na época trabalhavam no Massachussets Institute of Technology (M.I.T.). As letras RSA correspondem as iniciais de cada um dos inventores do código. Para o entendimento do funcionamento do método RSA serão abordados alguns tópicos de teoria dos números relacionados a alguns problemas: como encontrar números primos; como calcular os restos da divisão de uma potência por um número dado; os algoritmos de fatoração; o estudo da teoria dos números e a descrição do Método RSA, é o principal foco deste estudo.

REVISÃO BIBLIOGRÁFICA

Criptografia é um assunto que sofreu muitas mudanças durante a civilização humana, os homens inventavam códigos secretos na tentativa de transmitir mensagens inteligíveis por um interceptador.

O nome Criptografia, em grego, *cryptos* significa secreto, oculto e grafia significa escrita. A criptografia estuda os métodos para codificar uma mensagem de modo que só seu destinatário legítimo consiga interpretá-la (COUTINHO, 2011), ou seja, através dela é possível o envio de mensagens de uma forma segura, mesmo uma terceira pessoa tendo interceptado, não conseguirá ler a informação presente na mensagem.

Um dos primeiros fatos presentes no que concerne a Criptografia, foi utilizado por Júlio César para comunicar-se com as legiões em combate pela Europa. A criptografia era feita pela substituição de uma letra pela seguinte, isto é, transladava uma casa para diante.

Todo código necessariamente utiliza-se de duas propriedades, uma para codificar e

outra para decodificar. Decodificar é o que o destinatário faz quando recebe a mensagem e possui a chave de decodificação. Decifrar é tentar ler a mensagem sem saber a chave de decodificação.

A criptografia utilizada por Júlio César era muito fácil de ser decifrada, ou qualquer outra forma de criptografia utilizando apenas substituição de letras por outras, "isto se deve ao fato de que a frequência média com que cada letra é usada em uma língua é mais ou menos constante" (COUTINHO, 2011).

Uma maneira para contornar esse problema foi dividir a mensagem em grupos de letras e criptografar a mensagem em grupo, criando assim um sistema poligráfico, onde está presente as Cifras de Hill. Em 1929, Lester S. Hill publica seu livro *Cryptography in an Algebraic Alphabet*, no qual um bloco da mensagem é cifrado através de uma operação com matrizes.

O procedimento com as Cifras de Hill para criptografar uma mensagem era simples, bastava apenas saber as operações de Matrizes (Multiplicação e o cálculo da Inversa). O remetente e o destinatário sabiam a chave (matriz), que codificava a mensagem. Com isso, ao receber a mensagem conseguiam lê-la, caso uma terceira pessoa tivesse acesso a essa mensagem era preciso saber a chave que a codificou. Mas ainda essa forma de criptografar não era totalmente segura, pois se fosse descoberto o significado de uma coluna da matriz as demais poderiam ser descobertas.

Segundo Iezzi (2010), as matrizes surgiram na escola inglesa Trinity College, em um artigo do Matemático Arthur Cayley (1821-1895), datado de 1858. No século III a.C.; os chineses já desenvolviam um processo de resolução de sistemas lineares em que aparecia implícita a ideia das matrizes.

A utilização de matrizes foi fundamental para o desenvolvimento e agilidade na análise de dados. Segundo Dante (2010), quando você preenche um cadastro em uma página da *internet*, seus dados vão imediatamente para um banco de dados, que nada mais é do que uma matriz que relaciona as suas informações e de todos os outros cadastrados, às respectivas pessoas de forma coerente e recuperável.

A criptografia foi se aperfeiçoando durante os anos e em 1977 foi inventado um método criado por R.L. Rivest, A. Shamir e L. Adleman conhecido por RSA. "Há vários outros códigos de chave pública, mas o RSA é atualmente, o mais usado em aplicações comerciais" (COUTINHO, 2011).

O método RSA é muito seguro, utilizado no sistema bancário para garantir a segurança em transações financeiras pela internet. O método de codificação de uma

mensagem é muito simples, porém o processo inverso, o de decodificação é impossível de ser resolvido se a mensagem for interceptada por um hacker, mesmo utilizando-se da computação algébrica ou da programação computacional. Este fato de decifração da mensagem é um problema em aberto na Matemática até o presente momento, pois a única pessoa que consegue decodificar é o destinatário da mensagem. Segundo Coutinho (2011):

> *"para implementar o RSA precisamos de dois parâmetros básicos: dois números primos que vamos chamar de p e q. Para codificar uma mensagem usando o RSA é suficiente conhecer o produto dos dois primos, que vamos chamar de n. A chave de codificação do RSA é portanto constituída essencialmente pelo número n=pq. Cada usuário do método tem sua própria chave de codificação. Esta chave é tornada pública: todos ficam sabendo que, para mandar uma mensagem para o banco Acme, deve ser usada a chave n. Por isso n também é conhecido como "chave pública". Já a chave de decodificação é constituída pelos primos p e q. Cada usuário tem que manter sua chave de decodificação secreta ou a segurança do método estará comprometida".*

Aparentemente fatorar o número *n* parece ser um processo teoricamente fácil, mas usando como chaves de codificação RSA números muito grandes (de 200 algarismos ou mais), levaria milhares de anos. De acordo com Coutinho (2011), é disto que depende a segurança do RSA, da ineficiência dos métodos de fatoração atualmente conhecidos.

Na literatura pode-se encontrar alguns trabalhos sobre o estudo de criptografia com matrizes e criptografia RSA. Com relação as matrizes, Olgin (2011) comenta sobre "Criptografia para o desenvolvimento de atividades didáticas que aliem os conteúdos matemáticos do Ensino Médio a esse tema, que incentivem o manuseio de calculadoras científicas no Ensino de Matemática". Com relação a criptografia RSA, pelo fato da fatoração de $n = pq$ ser um problema em aberto na Matemática, há diversos pesquisadores que comentam sobre o assunto, entre eles Coutinho (2011).

Diante da importância do Método RSA utilizado constantemente nos dias atuais, o objetivo deste trabalha visa seu estudo, o porquê do Método ser seguro, além de propor uma alternativa para fatorar o valor de *n* por meio da função de Euler.

CAPÍTULO 1 - NÚMEROS PRIMOS

O objetivo deste capítulo é comentar sobre os diferentes tipos de funções que geram números primos, a relação dos números perfeitos com os primos de Merssene e a distribuição e crescimento dos números primos.

O estudo dos números primos é muito importante na Matemática, pois desempenham um papel fundamental que estão associados a muitos problemas famosos cujas soluções até o presente momento são desconhecidas, entre eles a criptografia RSA.

Um número natural maior que 1 e que só é divisível por 1 e por si próprio é chamado de *número primo.* Um número maior que 1 e que não é primo é denominado *composto*.

Não se conhece nenhuma fórmula que gere números primos arbitrariamente grandes.

Algumas fórmulas que produzem números primos são:

Fórmulas Polinomiais

$f(x) = x^2 - x + 41$, fornece números primos em sequência para $x = 0,1,2,...,40$, mas para $x = 41$, tem-se $f(41) = 41^2 - 41 + 41 = 41^2$, logo não é primo.

Fórmulas Exponenciais

$F_n = 2^{2^n} + 1$, chamado de primos de Fermat, são obtidos a partir de um número natural $n > 0$. Os primeiros quatro números de Fermat, obtidos pela função a partir de $n = 1,2,3,4$ são: $F_1 = 5, F_2 = 17, F_3 = 257, F_4 = 65537$. Em 1640, o matemático Pierre de Fermat observou que esses primeiros quatro números eram primos e conjecturou que todos os outros números naturais para $n \geq 5$, também seriam primos. Mas para $n = 5$, tem-se que $F_5 = 2^{2^5} + 1 = 4294967297$ e $4294967297 = 641 \cdot 6700417$, logo não é primo. Não se sabe se existe algum primo de Fermat para $n \geq 5$.

$M_p = 2^p - 1$, chamado de primos de Mersenne, são obtidos a partir de um número primo p. Nem todos os números $M_p = 2^p - 1$, com primo p, são primos. A Figura 1, mostra um exemplo programado no *software* WxMáxima, dos primeiros 200 números

primos aplicados à fórmula $M_p = 2^p - 1$, resultando em apenas 14 números primos.

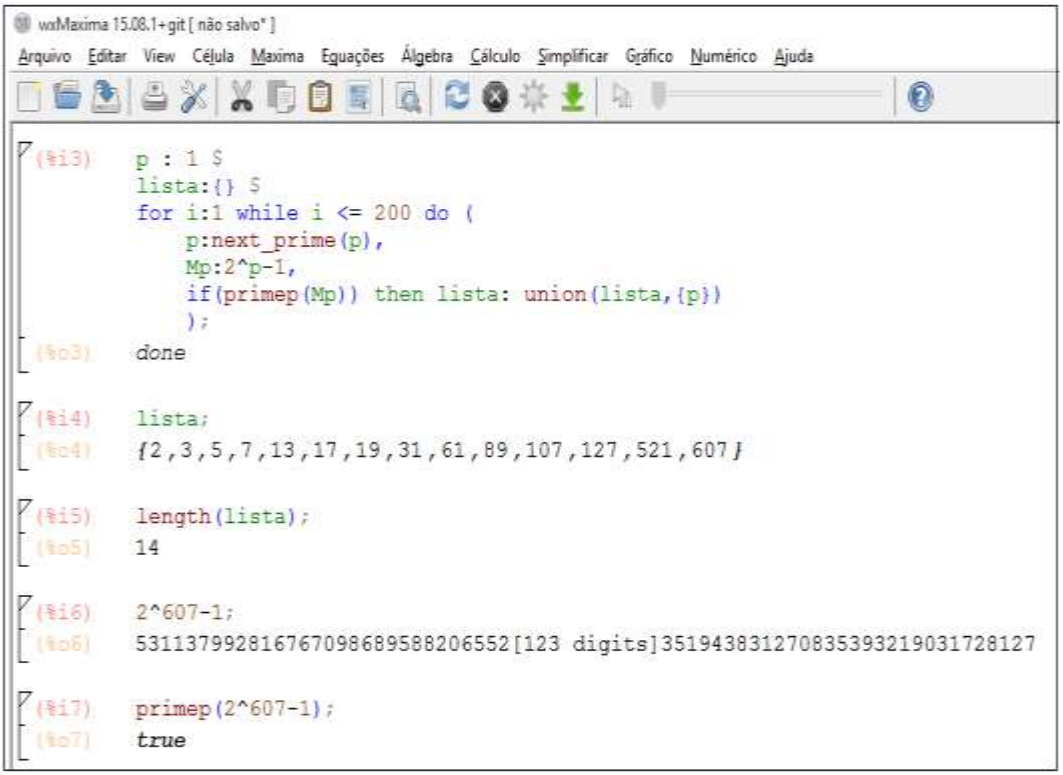

Figura1: Primos de Merssene

Fórmulas Fatoriais

Seja p primo. Define-se a função $p^{\#}$ como sendo a função obtida somente pelo produto de primos menores que ou iguais a p. Por exemplo, $3^{\#} = 2 \cdot 3 = 6$, se $q < p$ são primos sucessivos então. $p^{\#} = q^{\#} p$

Observe os números da forma $p^{\#} + 1$, na Tabela 1.

p	2	3	5	7	11	13
$p^{\#}$	2	6	30	210	2310	30030
$p^{\#} + 1$	3	7	31	211	2311	30031

Tabela 1: Aplicação da fórmula fatorial

Mas $13^{\#} + 1 = 30031 = 59 \cdot 509$, logo não gera um número primo.

1.1 NÚMEROS PERFEITOS

São os números naturais n com a seguinte propriedade: n é igual a soma de seus divisores próprios. Esses números fascinaram os gregos, a ponto de serem denominados de números perfeitos.

A função $\sigma(n)$, que calcula a soma de todos os divisores positivos de um número natural n, pode ser utilizada para o reconhecimento dos números perfeitos. Ao utilizar a função f que subtrai de cada número natural n a soma de seus divisores positivos próprios, ou seja, diferentes de 0 e do próprio número. Assim, esta função pode ser calculada da seguinte maneira:

$$f: N^* \to Z, \qquad n \to f(n) = n - [\sigma(n) - n] = 2n - \sigma(n)$$

A função f compara um número natural n com a soma de seus divisores próprios.

$$f(1) = 1 - [0] = 1$$
$$f(6) = 6 - [1 + 2 + 3] = 0$$
$$f(24) = 24 - [1 + 2 + 3 + 4 + 6 + 8 + 12] = -12$$
$$f(111) = 111 - [1 + 3 + 37] = 70$$

Os elementos do conjunto dos zeros da função f, são:

$$f^{-1}(0) = \{n \in N^* \mid f(n) = 0\} = \{6, 28, 496, 8128, 33550336, \ldots\}$$

Atualmente os elementos conhecidos desse conjunto são números pares e estes estão relacionados com os primos de Mersenne, por meio de um teorema devido parte a Euclides e parte a Euler.

Teorema 1.1.1. Um número natural n é um número perfeito par se, e somente se, $n = 2^{p-1} M_p$, onde M_p é um primo de Mersenne.

No WxMáxima o comando **divsum** calcula $\sigma(n)$, para $n \in N^*$, conforme Figura 2.

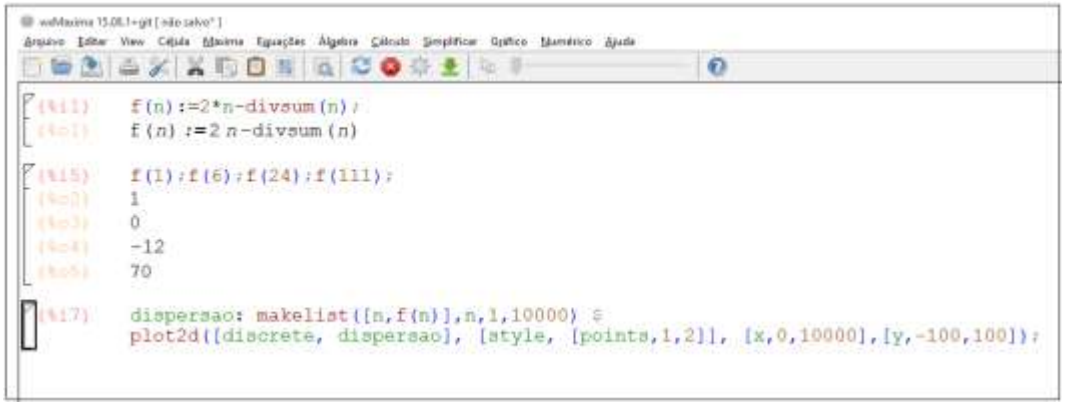

Figura 2: A função $f(n)=2n-\sigma(n)$ no WxMáxima.

Tem-se a dispersão dos pontos na Figura 3.

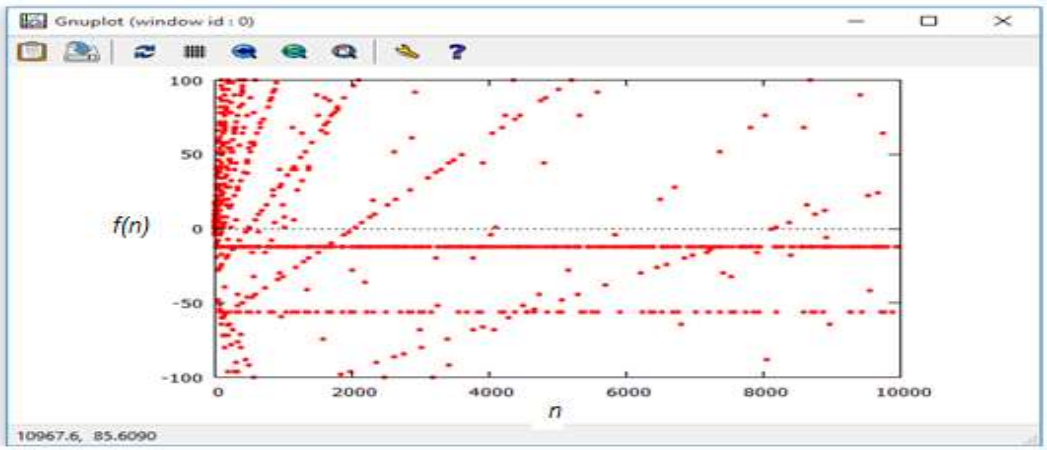

Figura 3: Gráfico da função $f(n)=2n-\sigma(n)$.

A dispersão horizontal dos pontos $(n, f(n))$ do gráfico da função $f(n)=2n-\sigma(n)$ se apresentam alinhados para os $f(n)=-12$ e $f(n)=-56$, mas isso se deve às seguintes proposições.

Proposição 1.1.1. Se $n=6p$ com p primo distinto de 2 e 3, então $f(n)=-12$.

Demonstração: Como p é um primo distinto de 2 e 3, segue que 6 e p não possuem divisores comuns além do 1. Logo, os divisores de $n=6p$ são: $1,2,3,6,p,2p,3p$ e $6p$. A soma desses divisores é $\sigma(n)=12+12p$ e $f(n)=2n-\sigma(n)=12p-(12+12p)=-12$.

Com o auxílio do WxMáxima pode-se listar alguns números com alinhamentos horizontais: $f(n)=-12$, $f^{-1}(-12)=\{n: f(n)=-12\}=\{24,30,42,54,66,78,102,114,138,174,\ldots\}$, conforme representado na Figura 4.

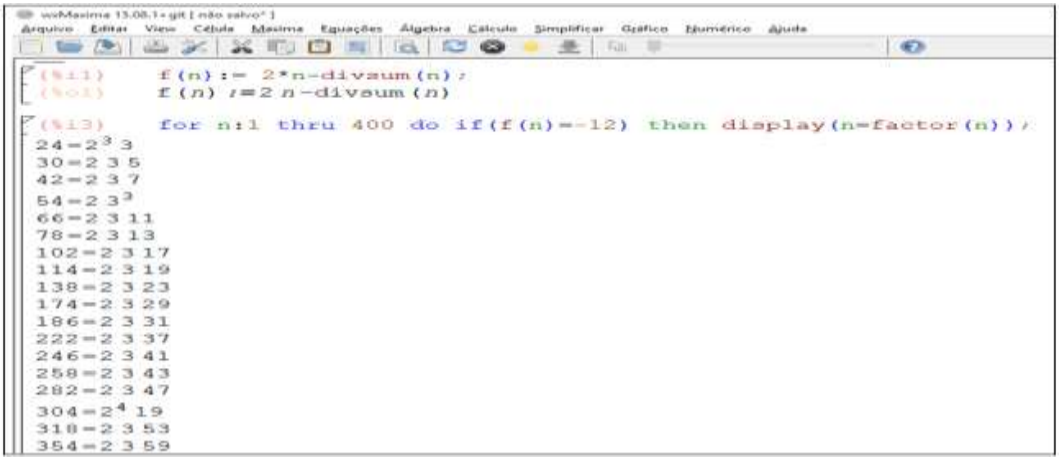

Figura 4: Valores para a função $f(n)=-12$.

Proposição 1.1.2. Se $n=28p$ com p primo distinto de 2 e 7, então $f(n)=-56$.

Demonstração: Como p é um primo distinto de 2 e 7, segue que 14 e p não possuem divisores comuns além do 1. Logo, os divisores de $n=28p$ são: $1,2,4,7,14,28,1p,2p,4p,7p,14p$ e $28p$. A soma desses divisores é $\sigma(n)=56+56p$ e $f(n)=2n-\sigma(n)=56p-(56+56p)=-56$.

Com o auxílio do WxMáxima pode-se listar alguns números com alinhamentos horizontais $f(n)=-56$, $f^{-1}(-56)=\{n: f(n)=-56\}=\{84,140,224,308,364,...\}$, conforme Figura 5.

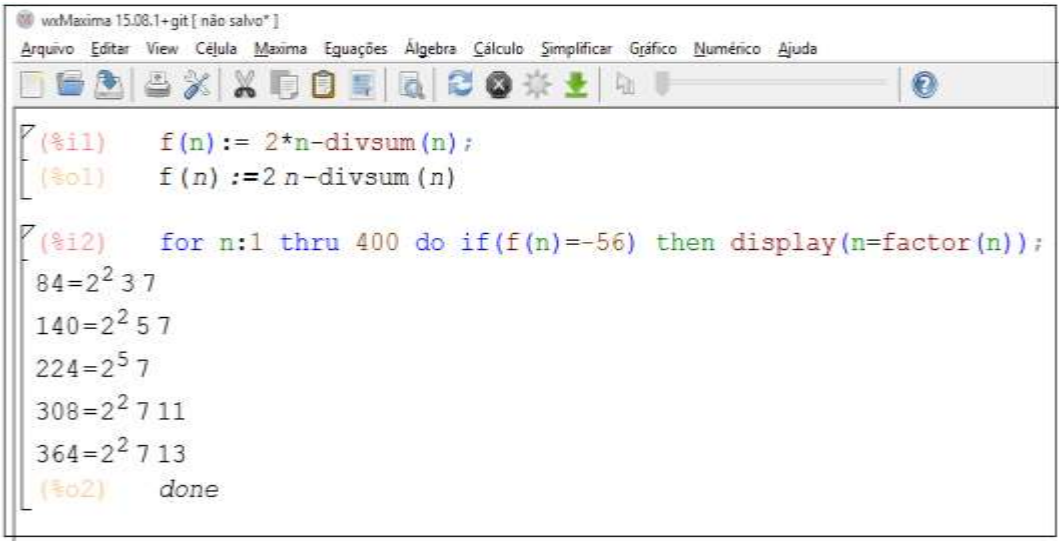

Figura 5: Valores para a função $f(n)=-56$.

A generalização para os demais números perfeitos se encontra na **Proposição 1.1.3.**

Proposição 1.1.3. Se K é um número perfeito e se $n=Kp$ com p primo não divisor de K, então $f(n)=-2\cdot K$

Demonstração: Como p é um primo distinto de K, segue que $K=k_1k_2...k_n$ e p não possuem divisores comuns além do 1. Logo, os divisores de $n=Kp$ são: $k_1,k_2,...,k_n,K,k_1p,k_2p,...,k_{n-1}p,Kp$ e k_np. A soma desses divisores é $\sigma(n)=2K+2Kp$ e $f(n)=2n-\sigma(n)=2Kp-(2K+2Kp)=-2K$.

1.2 A DISTRIBUIÇÃO DOS NÚMEROS PRIMOS

É possível estimar com boa aproximação, o número de primos inferiores a N, principalmente se N é grande, por outro lado a distribuição de números primos situados em pequenos intervalos tem comportamento aleatório. Para todo número $x > 0$, designa-se por $\pi(x)$ o número de primos p tais que $p \leq x$; $\pi(x)$ é chamada de função de contagem dos números primos.

Há questões a considerar, segundo Ribenboim (2012):

- O crescimento de $\pi(x)$: sua ordem de grandeza e a comparação de $\pi(x)$ com funções contínuas.

- Os resultados sobre o enésimo número primo, sobre a diferença entre dois: sua ordem de grandeza, sua regularidade ou sua irregularidade. Isso conclui a questão dos espaçamentos entre números primos consecutivos e conduz igualmente a um grande número de problemas em aberto, a saber:

- Os números primos em progressão aritmética.

- A conjectura de GOLDBACH.

- A distribuição dos números pseudoprimos e dos números de Carmichael.

Um número composto ímpar $n > 0$ é um número de Carmichael se $a^n \equiv a (\mathrm{mod}\, n)$ para todo $1 < a < n-1$. Portanto, números de Carmichael são pseudoprimos de Fermat para todas as bases. Em 1899, uma caracterização para os números de Carmichael foi dada por KORSELT.

Teorema 1.2.1. Um inteiro positivo ímpar *n* é um número de Carmichael se, e somente se, cada fator primo *p* de *n* satisfaz:

p^2 não divide *n* e $p-1$ divide $n-1$.

O número 561 é o menor número de Carmichael. Tem-se que: $561 = 3 \cdot 11 \cdot 17$, logo:

3^2 não divide 561, 11^2 não divide 561 e 17^2 não divide 561.

$3-1=2$ e 2 divide 560.

$11-1=10$ e 10 divide 560.

$17-1=16$ e 16 divide 560.

1.3 O CRESCIMENTO DE $\pi(x)$

Uma ideia no estudo da função $\pi(x)$ ou de outras funções ligadas à distribuição dos números primos é a comparação com funções clássicas que são calculáveis, cujos valores sejam próximos aos valores de $\pi(x)$. Considere $f(x)$ e $g(x)$ funções contínuas de valores reais positivos, definidas para $x \geq x_0 > 0$. $f(x) \sim g(x)$, significa que $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$ e então $f(x)$ e $g(x)$ são assintoticamente iguais, quando x tende para o infinito. Porém isso não significa que a diferença entre essa funções seja pequena, por exemplo, x^2 é assintótica a $x^2 - x$, mas a diferença entre elas cresce à medida que x tende ao infinito.

O Teorema dos Números Primos descreve a distribuição assintótica dos números primos, ou seja, como os primos estão distribuídos entre os números inteiros e $\frac{x}{\ln(x)}$ é uma boa aproximação para $\pi(x)$ uma vez que $\lim_{x\to\infty} \frac{\pi(x)}{\frac{x}{\ln(x)}} = 1$, ou seja, $\pi(x) \sim \frac{x}{\ln(x)}$.

A função $\frac{x}{\ln(x-a)}$, para qualquer constante real a, pode ser utilizada para aproximar $\pi(x)$. No Teorema dos Números Primos o valor de *a* é igual a zero, mas segundo alguns estudos Ribenboin (2012), conclui que $a = 1$ é a melhor escolha para a aproximação. Sendo assim, pode-se aproximar $\pi(x)$ utilizando-se a função $\frac{x}{\ln(x-1)}$.

CAPÍTULO 2 - TEOREMAS FUNDAMENTAIS

O objetivo deste capítulo é mencionar alguns teoremas importantes para o decorrer da pesquisa, sendo estes o Teorema Fundamental da Aritmética, Teorema da Fatoração Única, Teorema de Fermat e o Teorema de Wilson.

2.1 TEOREMA FUNDAMENTAL DA ARITMÉTICA

Todo número natural maior que 1 ou é primo ou se escreve de modo único (a menos da ordem dos fatores) como um produto de números primos.

Demonstração: Se $n = 2$, o resultado é obviamente verificado.

Suponha o resultado válido para todo número natural menor que *n*, tem-se que provar que vale para *n*. Se o número *n* é primo, nada a demonstrar. Suponha então, que *n* seja composto. Logo, existem números naturais n_1 e n_2 tais que $n = n_1 n_2$, com $1 < n_1 < n$ e $1 < n_2 < n$. Pela hipótese de indução, tem-se que existem números primos $p_1,...,p_r$ e $q_1,...,q_s$, tais que $n_1 = p_1...p_r$ e $n_2 = q_1...q_s$. Portanto, $n = p_1...p_r q_1...q_s$.

Para provar a unicidade da escrita, suponha, agora, que $n = p_1...p_r = q_1...q_s$, onde os p_i e os q_j são números primos. Se $p, p_1,...,p_n$ são números primos e, se $p \mid p_1...p_n$, então $p = p_i$ para algum $i = 1,...,n$. Como $p_1 \mid q_1...q_s$, tem-se que $p_1 = q_j$ para algum j, que, após o reordenamento de $q_1,...,q_s$, pode-se supor que seja q_1. Portanto, $p_2...p_r = q_2...q_s$. Como $p_2...p_r < n$, a hipótese de indução acarreta que $r = s$ e os p_i e q_j são iguais aos pares.

2.2 EXISTÊNCIA DA FATORAÇÃO

Para estudar a fatoração dos números primos é fundamental enunciar o Teorema da Fatoração Única.

Teorema 2.2.1 (**Teorema da Fatoração Única**)**.** Dado um inteiro positivo $n \geq 2$, pode-se sempre escrevê-lo, de modo único, na forma: $n = p_1^{e_1}...p_k^{e_k}$, em que

$1 < p_1 < p_2 < p_3 < ... < p_k$ são números primos e $e_1 ... e_k$ são inteiros positivos.

Este teorema encontra-se demonstrado mediante duas proposições (Prop. 30 e 32) dadas por Euclides no Livro VII de seus Elementos.

2.3 TEOREMA DE FERMAT

O Pequeno Teorema de Fermat afirma que se p é um número primo e a um inteiro qualquer então p divide $a^p - a$. Casos particulares desse teorema já eram conhecidos desde a antiguidade. Segundo Coutinho (2011), os chineses sabiam que se p é primo então p divide $2^p - 2$, mas foi Fermat quem obteve o resultado geral e o introduziu na Matemática européia do século XVII utilizando a linguagem de congruências.

Teorema de Fermat I. Seja p um número primo e a um inteiro, então $a^p \equiv a \pmod{p}$.
Demonstração: A prova deste teorema será feita por indução finita. Para isto precisa encontrar uma proposição $p(n)$ para aplicar a indução. $p(n)$: $n^p \equiv n \pmod{p}$.

É evidente que $p(1)$ é válido, pois $1^p = 1$. Suponha então que $n^p \equiv n \pmod{p}$. A passagem de $p(n)$ para $p(n+1)$ é pelo binômio de Newton. Para essa passagem utiliza-se o seguinte Lema 1.

Lema 1. Seja p um número primo e a e b inteiros. Então, $(a+b)^p \equiv a^p + b^p \pmod{p}$.
Demonstração do Lema 1: Utilizando a expressão usual do binômio de Newton, tem-se: $(a+b)^p = a^p + b^p + \sum_{i=1}^{p-1} \binom{p}{i} a^{p-i} b^i$. Para obter o lema é suficiente mostrar que o termo $\sum_{i=1}^{p-1} \binom{p}{i} a^{p-i} b^i$ é congruente a zero módulo p. Considere o número binomial $\binom{p}{i} = \frac{p(p-1)...(p-i+1)}{i!}$. Para que a fração corresponda a um número inteiro é preciso que o denominador seja completamente simplificado por termos no numerador. Suponha que $1 \le i \le p-1$, então o denominador $i!$ não tem p como um de seus fatores primos.

Assim o fator p que aparece no numerador não é cancelado por nenhum fator do denominador. Portanto o número inteiro $\binom{p}{i}$ é múltiplo de p quando $1 \le i \le p-1$, consequentemente $\sum_{i=1}^{p-1} \binom{p}{i} a^{p-i} b^i \equiv 0 \pmod{p}$.

Voltando à demonstração do Teorema de Fermat, supondo que $n^p \equiv n \pmod{p}$ e, deseja-se mostrar que $(n+1)^p \equiv n+1 \pmod{p}$. Utilizando o Lema 1, tem-se que:

$(n+1)^p \equiv n^p + 1^p \equiv n^p + 1 \pmod{p}$. Como a hipótese de indução é $n^p \equiv n \pmod{p}$, então, $(n+1)^p \equiv n^p + 1 \equiv n+1 \pmod{p}$. Com isso, prova-se o enunciado do teorema para os números naturais, mas este foi enunciado para qualquer inteiro. Então, chamando de a um inteiro negativo, $-a$ é positivo, logo aplicando o teorema já provado, para $-a$. Tem-se:

$$(-a)^p \equiv -a \pmod{p}. \qquad (2.1.)$$

Supondo que p é ímpar, $(-a)^p = -a^p$. Substituindo em (2.1.), $-a^p \equiv -a \pmod{p}$, e multiplicando por -1, concluí-se que $a^p \equiv a \pmod{p}$, restando apenas o caso em que $p = 2$. Se $p = 2$ então $(-a)^p = a^p$. O que recorre no teorema $a^p \equiv a \pmod{p}$.

Será utilizado a ideia do Pequeno Teorema de Fermat, no método RSA para justificar uma passagem Matemática na fórmula, mas sendo este o enunciado a seguir.

Teorema de Fermat II. Seja p um número primo e a um inteiro que não é divisível por p. Então $a^{p-1} \equiv 1 \pmod{p}$.

Demonstração: Segundo o Teorema de Fermat I, se p é primo e a é um número inteiro qualquer, então $a^p \equiv a \pmod{p}$. Suponha que p não divide a. Neste caso a é invertível módulo p, de acordo com o teorema de inversão. Seja a' um inteiro positivo tal que $aa' \equiv 1 \pmod{p}$. Multiplicando ambos os membros de $a^p \equiv a \pmod{p}$ por a', obtém-se:

$a' a a^{p-1} \equiv a' a \pmod{p}$. Com a substituição de $aa' \equiv 1 \pmod{p}$ nesta equação, tem-se $a^{p-1} \equiv 1 \pmod{p}$, o que conclui a demonstração.

Um exemplo da aplicação direta do Teorema de Fermat é dado a seguir.

Sejam a, k e p três inteiros positivos, dos quais sabe-se que p é primo e não divide a. Seja k um número muito grande; deseja-se encontrar a forma reduzida de a^k módulo p. Basta apenas que o valor de $k \geq p-1$, pois dividindo k por $p-1$, obtém-se $k = (p-1)q + r$, em que o resto r satisfaz $0 \leq r \leq p-2$. Tem-se então que:

$a^k \equiv a^{(p-1)q+r} \equiv (a^{p-1})^q a^r \pmod{p}$, mas pelo Teorema de Fermat II, tem-se que $a^{p-1} \equiv 1 \pmod{p}$. Então $a^k \equiv 1^q a^r \equiv a^r \pmod{p}$.

Se desejar calcular $2^{5432675}$ módulo 13. Utiliza-se desta forma, o Teorema de Fermat para calcular o resto da divisão de 5432675 por 12, resultando em 11. Assim:

$2^{5432675} \equiv 2^{11} \equiv 7 \pmod{13}$. Logo 7 é o menor resíduo positivo.

2.4 TEOREMA DE WILSON (T.W.)

Teorema 2.4.1. p é um número primo se, e somente se, $(p-1)! \equiv -1 \pmod{p}$.

Demonstração: $(\Rightarrow)$ Se p é primo, então todo elemento de Z_p, exceto [-1] e [1], possui um único inverso distinto de si. Logo:

$(p-2)\cdot(p-3)...3\cdot 2 \equiv 1 \pmod{p}$, mas $(p-1)! = (p-1)\cdot(p-2)\cdot...\cdot 2\cdot 1 \equiv p-1 \equiv -1 \pmod{p}$.

$(\Leftarrow)$ Suponha por absurdo que m seja composto. Então existe um inteiro d, com $1 < d < m$, que divide m. Portanto, $(m-1)! \equiv -1 \pmod{d}$. Por outro lado, como $d < m$, d é um divisor de $(m-1)!$ e $(m-1)! \equiv 0 \pmod{d}$, o que é uma contradição. Portanto, m é primo, o que conclui a demonstração do teorema.

Um exemplo da aplicação do Teorema de Wilson. Encontrar o menor resíduo positivo de $(8\cdot 9\cdot 10\cdot 11\cdot 12\cdot 13) \bmod 7$.

É fácil observar que:

$8 \equiv 1 \pmod{7}$
$9 \equiv 2 \pmod{7}$
$10 \equiv 3 \pmod{7}$
$11 \equiv 4 \pmod{7}$
$12 \equiv 5 \pmod{7}$
$13 \equiv 6 \pmod{7}$

Logo: $(8\cdot 9\cdot 10\cdot 11\cdot 12\cdot 13) \equiv 1\cdot 2\cdot 3\cdot 4\cdot 5\cdot 6 \pmod{7}$.

Pelo T.W., tem-se que $(p-1)! \equiv -1 \pmod{p}$, ou seja, $6! \equiv -1 \pmod{7}$.

Mas $6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$, logo:

Se $(8 \cdot 9 \cdot 10 \cdot 11 \cdot 12 \cdot 13) \equiv 6! (\mathrm{mod}\, 7)$ e $6! \equiv -1 (\mathrm{mod}\, 7)$, então $(8 \cdot 9 \cdot 10 \cdot 11 \cdot 12 \cdot 13) \equiv -1 (\mathrm{mod}\, 7)$.

Porém $6 \equiv -1 (\mathrm{mod}\, 7)$, logo $(8 \cdot 9 \cdot 10 \cdot 11 \cdot 12 \cdot 13) \equiv 6 (\mathrm{mod}\, 7)$.

Portanto, o menor resíduo positivo é igual a 6.

CAPÍTULO 3 - ALGORITMOS DE FATORAÇÃO

O objetivo deste capítulo é mencionar sobre alguns algoritmos de fatoração, a ineficiência dos algoritmos, a fatoração por Fermat e o algoritmo Euclidiano.

O processo de encontrar os fatores primos de um número composto denomina-se fatoração. Existem diversos algoritmos de fatoração, mas não existe um algoritmo que funcione perfeitamente, em que o computador possa executar em tempo polinomial para todos os números inteiros. Nesta pesquisa foram abordados alguns algoritmos de fatoração.

Considere o seguinte Problema: tendo por entrada o valor $n \in \mathrm{N}$, determine seus fatores primos e respectivos expoentes.

Com foco apenas no primeiro fator de um inteiro dado. Tendo *n* como entrada, tente dividir *n* por cada um dos inteiros de 2 a *n* – 1, caso algum desses inteiros dividir *n*, então encontra-se um fator de *n*, onde é o menor fator e este fator é um número primo.

Sabe-se que um número inteiro não pode ter um fator maior que ele próprio e também pode-se restringir a busca em um intervalo menor que 2 a *n* – 1, sendo este intervalo de 2 a $\sqrt{n}$. Porém se *n* é primo o único fator será o próprio *n*. É necessário verificar, entretanto, que se *n* é composto, seu menor fator é no máximo $\sqrt{n}$.

Assim, se *n* é um número composto e se $f > 1$ é seu menor fator, existe um inteiro positivo *a* tal que $n = f\,a$. Como *f* é o menor fator, certamente $f \le a$. Mas $a = \frac{n}{f}$, logo $f \le \frac{n}{f}$. Disso decorre que $f^2 \le n$, que é equivalente a $f \le \sqrt{n}$.

O procedimento descrito é representado pelo Algoritmo 3.1.

3.1 ALGORITMO DA FATORAÇÃO (MENOR FATOR)

Entrada: Digite um inteiro positivo *n*.
Saída : Inteiro positivo *f* que é o menor fator primo de *n* ou a indicação que *n* é primo.
Etapa1: Comece fazendo $f = 2$.

Etapa 2: Se $\frac{n}{f}$ é inteiro escreva ' *f* é fator de *n* ' e pare; senão vá para a Etapa 3.

Etapa 3: incremente a *f* uma unidade e vá para a Etapa 4.

Etapa 4: Se $f > \sqrt{n}$, escreva *n* é primo e pare. Caso contrário, retorne à Etapa 2.

Logo, dado um inteiro $n > 0$, pode-se determinar se *n* é primo ou composto. Se *n* é primo encontra-se a sua fatoração, mas se *n* for composto, pode-se encontrar todos os seus fatores primos e suas respectivas multiplicidades aplicando o algoritmo da fatoração várias vezes, ou seja, aplicando o algoritmo a *n* encontra-se o fator q_1. Então q_1 é o menor fator primo de *n*. Aplicando o algoritmo da fatoração ao co-fator $\frac{n}{q_1}$, determina-se um segundo fator q_2. Pode-se repetir esse procedimento aplicando ao co-fator $\frac{n}{q_1 q_2}$, e assim por diante. Dessa forma, determina-se uma sequência crescente de números primos $q_1 \leq q_2 \leq ... \leq q_s$, em que cada um é um fator de *n*.

O procedimento descrito é representado pelo Algoritmo 3.2.

3.2 ALGORITMO DOS FATORES (TODOS OS FATORES)

Entrada: Digite um inteiro positivo *n*.

Saída: $q_1, q_2, ..., q_s$, são os fatores primos de *n*, ou indicativo de que *n* é primo.

Etapa1: Comece fazendo $f = 2$.

Etapa 2: Se $\frac{n}{f}$ é inteiro armazenar $q_i = f$ e $n = \frac{n}{q_i}$. Vá para a Etapa 3.

Etapa 3: Se for verdade a etapa 2, então efetuar o cálculo para o novo valor em *n* e para o mesmo q_i para $i = 1,2,...,k$. Enquanto houver o mesmo fator repetir a Etapa 2. Senão vá para a Etapa 4.

Etapa 4: incremente a *f* uma unidade e vá para a Etapa 5.

Etapa 5: Se $f > \sqrt{n}$, então escreva os fatores $q_1, ..., q_s$ ou *n* é primo e pare. Senão volte a Etapa 2.

3.3 INEFICIÊNCIA DOS ALGORITMOS

Apesar da facilidade em entender e programar os algoritmos da fatoração e dos fatores, estes algoritmos são muitos ineficientes, mesmo com a tecnologia atual. O pior caso para executar o algoritmo é aquele em que o algoritmo executa o maior número de laços, ou seja, $\sqrt{n}$ laços. Para uma estimativa do tempo de execução, considere um número *n* primo, de 100 ou mais algarismos, ou seja, $n \geq 10^{100}$ e portanto o número de laços será igual a $\sqrt{n} \geq 10^{50}$. Assim, são necessárias pelo menos 10^{50} divisões para garantir que *n* é primo. Segundo Coutinho (2011), "Digamos que nosso computador executa 10^{10} divisões por segundo. Este é um número muito alto, que não é atingido no estado atual da tecnologia". Para estimar o tempo basta calcular $\frac{10^{50}}{10^{10}} = 10^{40}$ segundos para determinar que *n* é primo.

Um ano tem 60 (segundos) $\cdot$ 60 (minutos) $\cdot$ 24 (horas) $\cdot$ 365 (dias) $= 31536000$ segundos, resulta em $\frac{10^{40}}{31536000} = 3{,}1709791983\,764585 \cdot 10^{32}$ anos.

Portanto, percebe-se que é inviável confirmar que um número de 100 ou mais algarismos é primo usando esse algoritmo. Porém, isso também não significa que o algoritmo é inútil, segundo Coutinho (2011), "Se vamos fatorar um inteiro sobre o qual nada sabemos, há sempre a possibilidade que tenha um fator primo pequeno, digamos menor que 10^6", neste caso o algoritmo da fatoração pode ser utilizado.

Segundo Coutinho (2011), "É muito importante entender que não existe um algoritmo de fatoração que funcione bem para todos os inteiros: disso depende a segurança do método RSA". Ninguém sabe se a inexistência deste algoritmo geral é um problema intrínseco ou tecnológico, ou seja, se um tal algoritmo pode existir ou se ainda ninguém foi esperto o suficiente para inventá-lo.

3.4 FATORAÇÃO POR FERMAT

A fatoração por Fermat é muito eficiente quando *n* tem um fator primo próximo de $\sqrt{n}$. Supõe-se *n* ímpar, pois se *n* for par então 2 é um de seus fatores. Fermat teve a brilhante ideia de tentar encontrar inteiros positivos x e y tais que $n = x^2 - y^2$. Se

encontrados esses números $n = x^2 - y^2 = (x-y)(x+y)$ e por consequência, $x-y$ e $x+y$ são fatores de *n*.

Para implementar o algoritmo de Fermat primeiro é preciso determinar a parte inteira de $\sqrt{n}$. Se *n* é um quadrado perfeito então $f = \sqrt{n}$, será o próprio fator. Pela notação acima tem-se:

$x = f$ e $y = 0$. Para $y > 0$, então $x = \sqrt{n + y^2} > \sqrt{n}$. Logo pode-se elaborar o seguinte algoritmo.

Algoritmo de Fermat

Entrada: Inteiro positivo ímpar *n*.

Saída: Um fator de *n* ou uma mensagem que *n* é primo.

Etapa 1: Inicie $x = [\sqrt{n}]$; Se $n = x^2$, então x é fator de *n* e pode parar. Senão vá para a Etapa 2.

Etapa 2: Incremente x de uma unidade e calcule $y = \sqrt{x^2 - n}$.

Etapa 3: Repita a Etapa 2 até encontrar um valor inteiro para y (1° caso) , ou até que x seja igual a $\frac{n+1}{2}$ (2° caso): No 1° caso *n* tem fatores $x-y$ e $x+y$, no 2° caso *n* é primo.

Demonstração do Algoritmo de Fermat

É necessário considerar separadamente o que ocorre quando *n* é composto e quando *n* é primo. No caso de *n* ser composto, é necessário mostrar que existe um inteiro $x > [\sqrt{n}]$, em que os colchetes representa a parte inteira da raiz quadrada, tal que $\sqrt{x^2 - n}$ é um inteiro menor que $\frac{n+1}{2}$. Isto significa que se *n* é composto então o algoritmo irá parar antes de chegar em $\frac{n+1}{2}$. Se *n* é primo, então é necessário mostrar que o único valor possível para x é $\frac{n+1}{2}$.

Suponha que *n* pode ser fatorado na forma $n = pq$, em que $p \le q$. Deseja-se obter inteiros positivos x e y tais que $n = x^2 - y^2$, ou seja, $n = pq = (x-y)(x+y) = x^2 - y^2$.

Como $x - y \le x + y$, isto sugere que $p = x - y$ e $q = x + y$. Desse sistema, obtém-

se: $x = \frac{p+q}{2}$ e $y = \frac{q-p}{2}$, e portanto

$$\left(\frac{p+q}{2}\right)^2 - \left(\frac{q-p}{2}\right)^2 = \frac{p^2+2pq+q^2-q^2+2pq-p^2}{4} = pq = n. \quad (3.1.)$$

Entretanto, x e y devem ser números inteiros e por hipótese n é ímpar então $x = \frac{p+q}{2}$ e $y = \frac{q-p}{2}$, logo p e q, que são fatores de n, têm que ser ímpares. Com isso, $p+q$ e $q-p$ são pares e consequentemente, $\frac{p+q}{2}$ e $\frac{q-p}{2}$ são inteiros. Agora se n é primo então $p=1$ e $q=n$. E, $x = \frac{n+1}{2}$ é o único valor possível para x se n é primo. Resta agora considerar o caso em que n é composto. Se $p=q$, o algoritmo obtém a resposta na Etapa 1. Supondo que n é composto e não é um quadrado perfeito, isto é, $1 < p < q < n$. Neste caso, o algoritmo vai parar se forem satisfeitas as desigualdades: $[\sqrt{n}] \le \frac{p+q}{2} < \frac{n+1}{2}$.

A desigualdade da direita nos diz que $p+q < n+1$. Para $n = pq$, nesta última desigualdade, e subtraindo $q+1$ de ambos os membros, obtém-se $p-1 < q(p-1)$. Como $p > 1$, então $1 < p < q$. Logo $\frac{p+q}{2} < \frac{n+1}{2}$.

Para a desigualdade da esquerda, sabe-se que $[\sqrt{n}] \le \sqrt{n}$, e basta verificar que $\sqrt{n} \le \frac{p+q}{2}$. Logo esta desigualdade é válida se, e somente se, $n \le \frac{(p+q)^2}{4}$.

Pela Equação (3.1.), tem-se: $\left(\frac{p+q}{2}\right)^2 - \left(\frac{q-p}{2}\right)^2 = n$. Então $\frac{(p+q)^2}{4} - n = \frac{(q-p)^2}{4}$. Como $\frac{(q-p)^2}{4} \ge 0$, logo $\frac{(p+q)^2}{4} - n \ge 0$.

Este algoritmo de Fermat tem uma relação muito importante com a criptografia RSA, lembrando que a segurança do método RSA está na dificuldade em se fatorar a chave pública *n*, que é o produto de dois números primos. Pensar que escolher dois primos grandes basta para a segurança do método RSA é errôneo, pois se estes dois primos forem muito próximos, o seu produto irá gerar um número *n*, onde a sua raiz quadrada será próxima dos dois fatores primos, logo *n* é facilmente fatorável pelo algoritmo de Fermat.

3.5 ALGORITMO EUCLIDIANO

De acordo com Coutinho (2011), este algoritmo é descrito por Euclides nas Proposições 1 e 2 do Livro 7 dos Elementos de Euclides.

O objetivo do algoritmo Euclidiano é calcular o máximo divisor comum (MDC) entre dois números inteiros. Um inteiro b divide outro inteiro a, se existe um outro número inteiro c, tal que $a = bc$. Também diz que b é um divisor ou fator de a, ou ainda que a é múltiplo de b. O número c, definido acima, é denominado de co-fator de b em a. O MDC entre a e b é o maior inteiro positivo d que é divisor de a e também é divisor de b. Se d é o MDC entre a e b, escreve-se $d = MDC(a,b)$. Caso $MDC(a,b) = 1$, então os números são primos entre si ou co-primos.

Com a e b inteiros positivos e tais que $a \geq b$, o algoritmo Euclidiano consistem em dividir a por b, encontrando o resto r_1. Se $r_1 \neq 0$, dividindo b por r_1, obtém-se r_2. Se $r_2 \neq 0$, dividindo r_1 por r_2, obtém-se o resto r_3. O último resto diferente de zero, desta sequência de divisões é o máximo divisor comum (MDC) comum entre a e b. Para demonstrar o algoritmo Euclidiano, precisa-se do seguinte Lema 2.

Lema 2. Sejam a e b números inteiros positivos. Suponha que existam inteiro g e s tais que $a = bg + s$. Então $MDC(a,b) = MDC(b,s)$.

Demonstração: O lema diz que assumindo que a, b, g e s estão relacionados por $a = bg + s$ conclui-se que $MDC(a,b) = MDC(b,s)$. Para $d_1 = MDC(a,b)$ e $d_2 = MDC(b,s)$, tem-se que mostrar que $d_1 = d_2$. Então, basta mostrar que $d_1 \leq d_2$ e em seguida $d_2 \leq d_1$. Provando que $d_1 \leq d_2$.

Se $d_1 = MDC(a,b)$, então d_1 divide a e d_1 divide b. De acordo com a definição, isto significa que existem inteiros u e v tais que: $a = d_1 u$ e $b = d_1 v$. Substituindo na expressão $a = bg + s$, obtém-se: $d_1 u = d_1 vg + s$, ou seja, $s = d_1 u - d_1 vg = d_1(u - vg)$, logo d_1 divide s. Como o $d_1 = MDC(a,b)$, tem-se que d_1 divide b. Portanto d_1 é um divisor comum entre b e s, mas d_2 é o maior divisor comum entre b e s, logo $d_1 \leq d_2$. De modo análogo pode ser mostrado que $d_2 \leq d_1$ e consequentemente, $d_1 = d_2$.

Será utilizado o Lema 2, para provar que o último resto não nulo da sequência de

divisões é o MDC. Logo aplicando o algoritmo Euclidiano a a e b e supondo que o resto nulo ocorre após *n* divisões, tem-se:

$$
\begin{array}{lll}
a = bq_1 + r_1 & e & 0 \le r_1 < b \\
b = r_1 q_2 + r_2 & e & 0 \le r_2 < r_1 \\
r_1 = r_2 q_3 + r_3 & e & 0 \le r_3 < r_2 \\
r_2 = r_3 q_4 + r_4 & e & 0 \le r_4 < r_3 \\
\vdots & & \vdots \\
r_{n-4} = r_{n-3} q_{n-2} + r_{n-2} & e & 0 \le r_{n-2} < r_{n-3} \\
r_{n-3} = r_{n-2} q_{n-1} + r_{n-1} & e & 0 \le r_{n-1} < r_{n-2} \\
r_{n-2} = r_{n-1} q_n & e & r_n = 0
\end{array}
$$

Da última divisão tem-se que r_{n-1} divide r_{n-2}. Logo, o maior divisor comum entre os dois é o próprio r_{n-1}. Portanto $MDC(r_{n-2}, r_{n-1}) = r_{n-1}$. Com a aplicação do Lema 2 à penúltima divisão, conclui-se que $MDC(r_{n-3}, r_{n-2}) = MDC(r_{n-2}, r_{n-1}) = r_{n-1}$. E, com o Lema 2 sob a ante penúltima divisão, tem-se que $MDC(r_{n-4}, r_{n-3}) = MDC(r_{n-3}, r_{n-2}) = MDC(r_{n-2}, r_{n-1}) = r_{n-1}$. De modo análogo, conclui-se que o MDC(a,b) = r_{n-1}.

CAPÍTULO 4 - CRIPTOGRAFIA RSA

O objetivo deste capítulo trata-se da origem do Método RSA, descrição Matemática do método, pré-codificação, como codificar e decodificar os blocos, caso particular, o porquê do método RSA ser seguro, relacionar a função de Euler no método RSA, apontar as possibilidades de quebrar o método RSA e a escolha dos números primos.

Em 1976 Whitfield Diffie e Martin Hellman publicaram um documento denominado "As novas direções da criptografia", que sugeria o desenvolvimento de algum método para criptografar as informações antes de serem enviadas. Os dois cientistas propuseram um novo método para que a chave fosse enviada de forma segura, em que todas as informações necessárias eram disponibilizadas publicamente. A ideia consiste em usar uma função que seja fácil de calcular mas difícil de inverter computacionalmente, caso a pessoa não possua a chave do segredo. Essa função é chamada de "*função arapuca*" (*trap-door one-way function*).

Um código criptografado de chave pública deve conter um esquema público de codificação E e um esquema privado de decodificação D, em que E e D são fáceis de calcular e para uma mensagem M, $D(E(M))=E(D(M))=M$, ou seja, o procedimento de codificação E gera a mensagem codificado, em que o receptor de posse da chave de decodificação D, utilizando ela decodifique, resultando a mensagem original.

4.1 A ORIGEM DO MÉTODO RSA

Após a publicação do documento de Diffie e Hellman, três estudantes do Massachusetts Institute of Technology (MIT), começaram a pesquisar e desenvolver um novo tipo de criptografia, satisfazendo às condições estabelecidas no artigo. Para isso, eles estabeleceram um jogo de adivinhações, em que Rivest e Shamir comentavam algumas ideias de como criptografavam a mensagem e Adleman tentava adivinhar a técnica utilizada, mas certo dia Rivest trouxe um método que Adleman não conseguiu quebrar. Esse método então ficou conhecido por RSA em homenagem aos seus criadores (Ronald Rivest, Adi Shamir e Leonard Adleman), permanecendo inviolado até o presente momento.

É claro que durante esses anos, alguns pesquisadores encontraram fraquezas na implementação do método RSA, mas que foram corrigidas. Foram testadas várias chaves

RSA, propostas como desafio para analisar a escolha dos números primos e os métodos utilizados para encontrá-los. O RSA tornou-se a melhor maneira de criptografar as informações, como por exemplo, transações com cartão de crédito via internet.

O RSA é o resultado de dois cálculos matemáticos, um para codificar e outro para decodificar, em que se utilizam duas chaves criptográficas, uma chave pública e uma privada.

4.2 DESCRIÇÃO MATEMÁTICA DO MÉTODO

Para codificar uma mensagem, precisa-se de n, que é o produto de dois números primos p e q, logo $n = pq$ e de um inteiro positivo λ, que seja invertível módulo $\varphi(n)$, ou seja, o $MDC(\lambda, \varphi(n)) = 1$, em que $\varphi(n) = (p-1)(q-1)$. Denomina-se o par (n, λ) *chave de codificação* e o par (n, d) *chave de decodificação* do método RSA.

- Represente a mensagem com números inteiros, quebrando em bloco de maneira que esses blocos não ultrapassem o valor de n e não iniciem em zero.

- Para codificar a mensagem B, eleva-se cada bloco B_i à "$\lambda - ésima$" potência módulo n, ou seja, $B_i^{\lambda} \equiv A_i \pmod{n}$. Então cada resultado criptografado é o valor em A_i.

- Para decodificar a mensagem A criptografada, eleve-a a uma outra potência d e calcule o resto da divisão por n, ou seja, $A_i^{d} \equiv B_i \pmod{n}$. Então o resultado descriptografado é o valor em B.

O valor do d é o inverso de $\lambda[\mathrm{mod}\,(p-1)(q-1)] = \lambda[\mathrm{mod}\ \varphi(n)]$, ou seja, $\lambda d \equiv 1 (\mathrm{mod}\,\varphi(n))$

4.3 PRÉ-CODIFICAÇÃO

A primeira coisa a fazer para utilizar o método RSA é converter a mensagem em uma sequência de números relacionados em uma tabela de letras com seus respectivos números. Um cuidado na hora de relacionar as letras com os números é importante: por exemplo, se escolher a letra A = 1, B = 2 e assim por diante, quando escrever o número 12, qual será a interpretação? Nesse sistema de conversão há uma ambiguidade, se 12 = AB ou se 12 = L. Como alternativa, será utilizada a seguinte tabela de conversão de letras para

números (Tabela 2).

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

Tabela 2: Tabela de conversão.

Para aplicar o método RSA, será pré-codificado a palavra *RSA* pela Tabela 2. Com a conversão da mensagem letra a letra, obtém-se: $R=27, S=28, A=10$, assim a mensagem codifica é dada pelo número: 27.28.10.

4.4 CODIFICANDO E DECODIFICANDO

Serão utilizados números primos "pequenos" para fins pedagógicos, afim de que os cálculos possam ser verificados facilmente. Porém para uma maior segurança recomenda-se fatores grandes diante da dificuldade da fatoração.

Considere $p=5$ e $q=7$, tem-se que $n=pq=5\cdot 7=35$ e $\varphi(n)=(p-1)(q-1)$, logo $\varphi(n)=24$.Para iniciar o processo deve-se quebrar o código 27.28.10 em blocos menores que $n=35$.

B1	B2	B3
27	28	10

Tabela 3: Código em blocos.

Cada um dos blocos será codificado por $B_i^{\lambda}\equiv A_i \pmod n$, em que o $MDC(\lambda,\varphi(n))=1$, então $MDC(\lambda,24)=1$, escolhendo assim $\lambda=7$. Logo, codifica-se da seguinte maneira:

$B_i^{\lambda}\equiv A_i \pmod n$

$27^7\equiv A_1 \pmod{35}$ $27^7\equiv(-8)^7\equiv[(-8)^2]^3\cdot(-8)\equiv 64^3\cdot(-8)\equiv(-6)^3\cdot(-8)\equiv 48\equiv 13 \pmod{35}$

$28^7\equiv A_2 \pmod{35}$

$28^7 \equiv (-7)^7 \equiv [(-7)^2]^3 \cdot (-7) \equiv 49^3 \cdot (-7) \equiv 14^3 \cdot (-7) \equiv 21 \cdot (-3) \equiv 7 \pmod{35}$

$10^7 \equiv A_3 \pmod{35}$

$10^7 \equiv (10^2)^3 \cdot 10 \equiv (-5)^3 \cdot 10 \equiv (-125) \cdot 10 \equiv 15 \cdot 10 \equiv 10 \pmod{35}$

Os valores são: $A_1 = 13, A_2 = 7$ e $A_3 = 10$, logo a mensagem codificada é: 13.7.10

Mensagem Original	27.28.10
Mensagem Codificada	13.7.10

Tabela 4: Código de conversão.

A informação necessária para a decodificação consiste no par (n, d), lembrando que $\varphi(n) = (p-1)(q-1)$ e o valor d é o inverso de λ em $\varphi(n)$, ou seja, $\lambda d \equiv 1 \pmod{\varphi(n)}$. Neste caso, $\lambda = 7$, $\varphi(n) = 24$, donde $7d \equiv 1 \pmod{24}$, logo $d = 7$. Assim, decodifica-se 13.7.10, da seguinte maneira:

$A_i^d \equiv B_i \pmod{n}$

$13^7 \equiv B_1 \pmod{35}$

$13^7 \equiv (13^2)^3 \cdot (13) \equiv (-6)^2(-6) \cdot 13 \equiv (-6) \cdot 13 \equiv -78 \equiv 27 \pmod{35}$

$7^7 \equiv B_2 \pmod{35}$

$7^7 \equiv (7^2)^3 \cdot 7 \equiv 14^3 \cdot 7 \equiv 196 \cdot 14 \cdot 7 \equiv -147 \equiv 28 \pmod{35}$

$10^7 \equiv B_3 \pmod{35}$

$10^7 \equiv (10^2)^3 \cdot 10 \equiv (-5)^3 \cdot 10 \equiv (-125) \cdot 10 \equiv 15 \cdot 10 \equiv 10 \pmod{35}$

Os valores são: $B_1 = 27, B_2 = 28$ e $B_3 = 10$, logo a mensagem original é: 27.28.10, que corresponde às letras RSA.

4.5 UM CASO PARTICULAR DO RSA

Escolhendo números primos p e q da seguinte maneira:

$p \equiv 5 \pmod{6}$ e $q \equiv 5 \pmod{6}$, logo tem-se que $p - 1 \equiv 4 \pmod{6}$ e $q - 1 \equiv 4 \pmod{6}$, então $(p-1)(q-1) \equiv 16 \equiv 4 \pmod{6}$, pode-se escrever $(p-1)(q-1) = 6k + 4 = 6k + 3 + 1$, ou seja, $(p-1)(q-1) = 3(2k+1) + 1$, logo:

$3(2k+1) \equiv -1 \pmod{6k+4}$, multiplicando por -1, para que o resto seja 1.

$3(-2k-1) \equiv 1 \pmod{6k+4}$, somando $6k+4$, obtém-se:

$3(4k+3) \equiv 1 \pmod{6k+4}$.

Por exemplo, considere $p=5$ e $q=11$, logo $n=5\cdot 11=55$ e $\varphi(n)=(p-1)(q-1)=(5-1)(11-1)=40$, lembrando que, para codificar, utiliza-se o par (n,λ), em que $MDC(\lambda,\varphi(n))=1$. Neste caso, como p e q estão no caso particular, pois ambos deixam resto 5 na divisão por 6, então pode ser utilizado $\lambda=3$, mas por que esse valor para λ? Na sequência será respondida essa pergunta, mas antes, para decodificar, precisa-se conhecer o par (n,d).

O valor de d é calculado por $\lambda d \equiv 1 (\mod \varphi(n))$. Como $\lambda=3$, fazendo uma relação com o caso particular desenvolvido, em que $3(4k+3)=1 \pmod{6k+4}$, tem-se que $d=4k+3$ e $\varphi(n)=6k+4$. Esta relação justifica o porquê de $\lambda=3$. Nesse exemplo, pode ser calculado o valor de k em $\varphi(n)=6k+4$, pois sabe-se o valor de $\varphi(n)=40$, assim $40=6k+4 \Rightarrow k=6$. De posse do valor de $k=6$, calcula-se o valor $d=4k+3$, então $d=4\cdot 6+3=27$. Portanto é fácil calcular o valor de d, conhecendo-se o $\varphi(n)$, sem precisar aplicar o algoritmo de Euclides.

4.6 POR QUE O CASO PARTICULAR FUNCIONA?

O método RSA só será útil se, decodificando os blocos codificados, obtém-se novamente o bloco correspondente da mensagem original. Considere que os blocos estejam no intervalo de $1 \le B < n$ e $B^3 \equiv A (\mod n)$ para codificar, em que $0 \le A < n$, logo A é a codificação do bloco B. Em seguida, para decodificar utiliza-se $A^d \equiv B (\mod n)$.

Supondo que para decodificar $A^d \equiv e (\mod n)$, em que $0 \le e < n$, assim e é a decodificação do bloco A. Então, $e \equiv A^d \equiv (B^3)^d (\mod n)$, ou seja, $e \equiv B^{3d} (\mod n)$, mas $3d \equiv 1 (\mod (p-1)(q-1))$, ou seja, $3d \equiv 1 + k(p-1)(q-1)$, e tem-se: $B^{3d} \equiv B^{1+k(p-1)(q-1)} \equiv B \cdot B^{k(p-1)(q-1)}$. Basta provar que $B^{3d} \equiv B (\mod p)$ e $B^{3d} \equiv B (\mod q)$. Tem-se dois casos a considerar ($MDC(p,B) \neq 1$ e $MDC(p,B)=1$).

Se $MDC(p,B) \neq 1$, como p é um número primo, logo B será múltiplo de p, então $B=\lambda p$, ou seja, $B \equiv 0 (\mod p)$, logo $B^{3d} \equiv B (\mod p)$.

Se $MDC(p,B)=1$, $B^{3d} = B \cdot B^{k(p-1)(q-1)} = B \cdot (B^{p-1})^{k(q-1)}$, pelo Teorema de Fermat

tem-se que $B^{p-1} \equiv 1 (\text{mod } p)$ e $B \cdot (B^{p-1})^{k(q-1)} \equiv B \cdot 1^{k(q-1)} \equiv B (\text{mod } p)$. De modo análogo demonstra-se que $B^3 \equiv B (\text{mod } q)$. Considere o seguinte sistema de congruências:

$$\begin{array}{l} x \equiv B (\text{mod } p) \quad \Rightarrow \; x = B + t_1 p \; \Rightarrow \; x - B = t_1 p \\ x \equiv B (\text{mod } q) \quad \Rightarrow \; x = B + t_2 q \; \Rightarrow \; x - B = t_2 q \end{array}, \text{ então:}$$

$x - B = t_3 pq \Rightarrow x \equiv B (\text{mod } pq) \Rightarrow x \equiv B (\text{mod } n)$ e $B^{3d} \equiv B (\text{mod } p)$, como $0 \leq e < n$ e $1 \leq B < n$, então necessariamente a congruência implica a igualdade, portanto concluí-se que $e = b$.

4.7 POR QUE O RSA É SEGURO?

Cabe ressaltar que o RSA é um método de chave pública, então sejam p e q os dois números primos do método, e $n = pq$. A chave de codificação corresponde à chave pública. Portanto o par (n, λ) é acessível para qualquer usuário. O método RSA só será seguro se for difícil calcular d, quando apenas se conhece os valores de (n, λ).

Para calcular o valor de d, aplica-se o algoritmo Euclidiano estendido a $\varphi(n)$ e λ, pois $\lambda d \equiv 1 (\text{mod } \varphi(n))$. Mas para se calcular $\varphi(n)$, é necessário saber quais são os primos p e q, pois $\varphi(n) = (p-1) \cdot (q-1)$. Portanto para decifrar o código precisa-se fatorar n. Porém se n é um número muito grande, sabe-se que fatorar n é um problema extremamente difícil, pois não se conhece um algoritmo rápido de fatoração.

4.8 ANÁLISE DA FUNÇÃO DE EULER NO MÉTODO RSA

Suponha que $n = pq$ e $\varphi(n) = (p-1) \cdot (q-1)$ sejam ambos conhecidos. Pode-se determinar p e q a partir deles.

Para $\varphi(n) = (p-1) \cdot (q-1)$, tem-se que $\varphi(n) = pq - (p+q) + 1 = n - (p+q) + 1$, logo $p + q = n - \varphi(n) + 1$, portanto tem-se a soma dos dois números primos.

Sabe-se que $(p+q)^2 - 4n = (p^2 + 2pq + q^2) - 4n = (p^2 + 2pq + q^2) - 4pq = (p-q)^2$, logo:

$p - q = \sqrt{(p+q)^2 - 4n}$ ou $p - q = \sqrt{(n - \varphi(n) + 1)^2 - 4n}$, portanto tem-se a diferença dos dois números primos.

Sendo conhecidos $p+q$ e $p-q$, calcula-se facilmente o valor de p e q por meio da resolução de um sistema linear, ou seja, o procedimento fatora o valor em n.

4.9 A FUNÇÃO $f(m)=(m-n-1)^2-4n$

Seja a função $f:\ R\to R$, tal que $f(m)=(m-n-1)^2-4n$. Assim, tem-se ainda que $f(m)=m^2-2mn-2m+n^2-2n+1$, cujas raízes são: $m_1=n+1+2\sqrt{n}$ e $m_2=n+1-2\sqrt{n}$. A raiz m_1 é positiva, basta analisar m_2. Logo, $m_2=n+1-2\sqrt{n}>0$, tem-se $n+1>2\sqrt{n}$. Elevando ao quadrado ambos os membros obtém-se $(n+1)^2>(2\sqrt{n})^2$. Logo $n^2+2n+1>4n$ que é equivalente a $n^2-2n+1>0$ e $n^2-2n+1=(n-1)^2>0$; portanto ambas as raízes são positivas.

Escolher dois números primos quaisquer $p<q$, efetuar o produto deles para gerar o valor de n e efetuar a subtração de q por p para gerar o valor de g, logo $n=pq$ e $g=q-p$. Considere a função $f:\ R\to R$, tal que $f(m)=(m-n-1)^2-4n$. Substituir o resultado de g^2 no lugar de $f(m)$ e resolver a equação em função de m. Sendo assim, $g^2=(m-n-1)^2-4n$ ou $m^2-2mpq-2m+p^2q^2+1-q^2-p^2=0$. As raízes dessa equação são $m_1=1-(p+q)+pq$ e $m_2=1+p+q+pq$. O importante para o método RSA é apenas a menor raiz, sendo esta o $m_1=1-(p+q)+pq=(p-1)\cdot(q-1)$.

O que é notável, caso alguém invente um método rápido que encontre o valor de $g^2=(q-p)^2$, fica fácil de encontrar os fatores primos p e q.

O conceito do vértice da parábola por meio das raízes é a média aritmética das raízes que resulta no x_v da função, ou seja, $x_v=\dfrac{1-(p+q)+pq+1+p+q+pq}{2}=\dfrac{2pq+2}{2}=pq+1$ e também o conceito do y_v, calculado por meio de $f(x_v)$, ou seja, $f(x_v)=(pq+1-pq-1)^2-4pq=-4pq$.

Na figura 6, tem um esboço do gráfico da função $f(m)=(m-n-1)^2-4n$, que relaciona o vértice da parábola com o valor procurado de $m_2=n+1-2\sqrt{n}$ que é a soma dos números primos *p* e *q*.

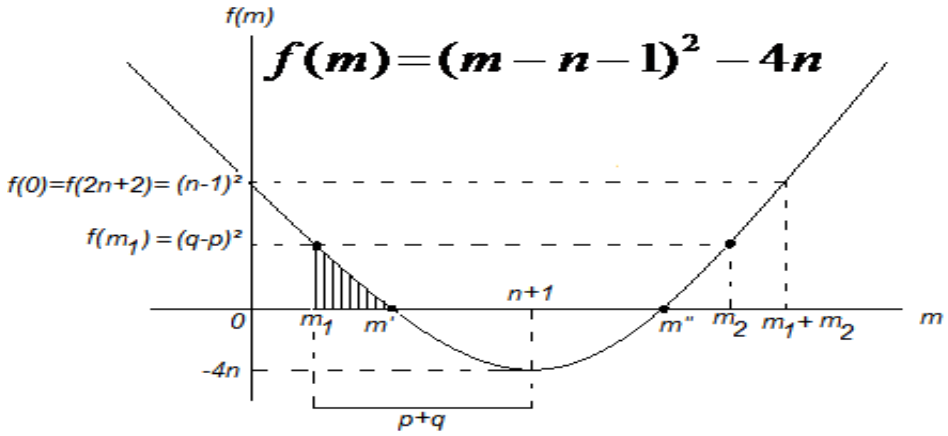

Figura 6: A função $f(m)=(m-n-1)^2-4n$

4.10 A ESCOLHA DOS NÚMEROS PRIMOS

Um ponto muito importante no método RSA é a escolha dos dois números primos p e q que serão utilizados, pois se ambos forem muito pequenos é fácil de encontrá-los.Entretanto não basta que ambos sejam muito grandes para garantir a segurança do método, pois se $|p-q|$ é pequeno, é fácil fatorar $n=pq$, utilizando o algoritmo de Fermat.

Segundo Coutinho (2011), em 1995 dois estudantes de uma universidade americana, quebraram a versão do RSA em uso público, pelo fato da escolha dos números primos ser totalmente inadequada.

Para implementar o RSA com chave pública (n,λ), de modo que n tenha r algarismos sugere-se que o primo p esteja no intervalo de $\frac{4r}{10}$ e $\frac{45r}{100}$ algarismos e em seguida, supor que o primo q seja próximo de $\frac{10^r}{p}$, (COUTINHO, 2011).

4.11 UMA ANÁLISE PARA QUEBRAR O RSA

Na área de estudos em algoritmos de fatoração não é conhecido um algoritmo determinístico, em tempo polinomial, que encontre um fator de um inteiro composto *n* com muitos algarismos. Este fato está diretamente relacionado ao método RSA, pois caso este algoritmo existisse, seria possível encontrar facilmente os fatores primos de *n*, e

consequentemente saber o valor de $\varphi(n)$ e também encontrar o expoente d, que é a chave de decodificação do par (n,d).

Nota-se algumas possibilidades de decifrar o método RSA, sendo apenas conhecido o valor de *n*: fatorar *n*, encontrar $\varphi(n)$ ou encontrar *d* sem fatorar *n* ou encontrar $\varphi(n)$.

Para o caso de encontrar $\varphi(n)$, conhecendo apenas *n*, utilizando a função $f(m)=(m-n-1)^2-4n$, no apêndice encontra-se o algoritmo implementado no Maple 12 que calcula o $\varphi(n)$. Observa-se que a busca pelo valor de $\varphi(n)$, pode ser feita por meio da menor raiz na função, sendo verificado se a parte inteira da raiz é múltiplo de quatro; caso negativo, deve-se subtrair um até encontrar o primeiro valor múltiplo de quatro. Substitui-se esse valor próximo da raiz, se o resultado for um quadrado perfeito, então o valor é o $\varphi(n)$; senão subtrai-se quatro e substitui novamente na função até gerar um quadrado perfeito.

O fato de verificar se o número é um múltiplo de quatro, e também de subtrair quatro até encontrar um quadrado perfeito, deve-se a característica da função $\varphi(n)=(p-1)\cdot(q-1)$, pois *p* e *q* são primos, logo $\varphi(n)$ será o produto de dois números pares, resultando num múltiplo de quatro.

Portanto, se existisse um método aplicado na função $f(m)=(m-n-1)^2-4n$, que encontrasse rapidamente um quadrado perfeito no intervalo de 0 a $n+1-2\sqrt{n}$, então o valor aplicado na função seria o valor de $\varphi(n)$.

CAPÍTULO 5 - APLICAÇÕES EM SALA DE AULA

O objetivo deste capítulo é propor atividades motivadoras que possam auxiliar os professores no momento de ensinar determinados conteúdos.

5.1 CRIPTOGRAFIA RSA REDUZIDA

Tema: Criptografia RSA Reduzida

Objetivo: Essa atividade propõe o estudo do método RSA apenas utilizando-se de um número primo para o valor em $n = p$ e a função $\varphi(n) = p - 1$.

Conteúdos Relacionados: Números Primos, Potenciação, Congruências, Divisão de Números Naturais.

Série de Aplicação: 9º ano

Duração: 6 aulas

Recursos Pedagógicos: Lousa e giz, caderno, lápis e calculadora.

Metodologia: Os alunos deverão sentar em duplas e um deles irá criptografar uma mensagem e o outro irá tentar decodificá-la. Será utilizado a chave $n = p$ com (p primo) , $p \equiv 5 \pmod 6$ e a função $\varphi(n) = p - 1$ e $\lambda = 3$. Como exemplo será apresentado aos alunos a palavra “MESTRE”, utilizando como chave de codificação o par $n = 17$ e $\lambda = 3$. O primeiro passo é relacionar a mensagem com os números, como na Tabela 2: Tabela de conversão, que consta na seção 4.3. Sendo assim, a mensagem a ser criptografada é 22.14.28.29.27.14 , logo quebrando a mensagem em blocos menores que 17, tem-se:

2	2	14	2	8	2	9	2	7	14
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10

Tabela 5: Código em blocos

Cada um desses blocos será criptografado por $B_i^{\lambda} \equiv A_i \pmod{n}$. Durante a codificação será explicado a relação do resto das divisões e a importância da congruência, utilizando a calculadora, lousa e giz.

Calculando $B_i^{\lambda} \equiv A_i \pmod{n}$, tem-se:

$2^3 \equiv 8 \pmod{17}$

$2^3 \equiv 8 \pmod{17}$

$14^3 \equiv 7 \pmod{17}$

$2^3 \equiv 8 \pmod{17}$

$8^3 \equiv 2 \pmod{17}$

$2^3 \equiv 8 \pmod{17}$

$9^3 \equiv 15 \pmod{17}$

$2^3 \equiv 8 \pmod{17}$

$7^3 \equiv 3 \pmod{17}$

$14^3 \equiv 7 \pmod{17}$

Sendo assim, a mensagem codificada é 8.8.7.8.2.8.15.8.3.7 .

A informação necessária para poder decodificar consiste no par (n,d). Em que $\varphi(n) = p-1$, e o valor do d é o inverso de λ em $\varphi(n)$, ou seja, $\lambda d \equiv 1 \pmod{\varphi(n)}$, logo $3d \equiv 1 \pmod{16}$, portanto d = 11. Pelo Capítulo 4.5, se $p \equiv 5 \pmod{6}$ e $q \equiv 5 \pmod{6}$ então $d = 4k+3$ e $\varphi(n) = 6k+4$. Nesse caso, tem-se que $p \equiv 5 \pmod{6}$, pois $p = 17$. Como $\varphi(n) = 6k+4$ e $16 = 6k+4$, então $k = 2$. Assim $d = 4k+3$, e $d = 4 \cdot 2 + 3 = 11$.

Logo para a decodificação será explicado que $\varphi(n) = 6k+4$ e $d = 4k+3$, ficando a cargo do aluno que irá decodificar, efetuar os cálculos e encontrar o valor de d.

Para decodificar utiliza-se $A_i^{d} \equiv B_i \pmod{n}$, logo:

$8^{11} \equiv 2 \pmod{17}$

$8^{11} \equiv 2 \pmod{17}$

$7^{11} \equiv 14 \pmod{17}$

$8^{11} \equiv 2 \pmod{17}$

$2^{11} \equiv 8 \pmod{17}$

$8^{11} \equiv 2 \pmod{17}$

$15^{11} \equiv 9 \pmod{17}$

$8^{11} \equiv 2 \pmod{17}$

$3^{11} \equiv 7 \pmod{17}$

$7^{11} \equiv 14 \pmod{17}$

Sendo assim a mensagem decodificada é 2.2.14.2.8.2.9.2.7.14. Como $n = 17$, basta unir os blocos com dois algarismos cada, e retornando nos números: 22.14.28.29.27.14 e a mensagem é decodificada.

Após todos codificarem e decodificarem as mensagens, será proposta uma mensagem pelo professor, para averiguar se todos conseguiram alcançar o objetivo da atividade.

Avaliação: A avaliação será da seguinte maneira: o aluno que codificou corretamente a mensagem terá pontuação máxima. O aluno que decodificou corretamente terá pontuação máxima; o aluno que decodificou a mensagem do professor corretamente terá a pontuação máxima na atividade e caso o aluno que codificou erre no momento da codificação esse será auxiliado pelo professor para que possa alcançar o objetivo; assim, como o aluno que errou no momento de decodificar, ou no momento de calcular o valor de d .

5.2 CRIPTOGRAFIA RSA E A FUNÇÃO DO SEGUNDO GRAU

Tema: Criptografia RSA e a Função do Segundo Grau.

Objetivo: Essa atividade propõe o estudo da Função do Segundo Grau ,utilizando-se do método RSA, para motivar o aprendizado, especificamente no cálculo das raízes.

Conteúdos Relacionados: Função do Segundo Grau, Cálculo do Discriminante, Cálculo das Raízes e Números Primos.

Série de Aplicação: 1º ano do Ensino Médio

Duração: 6 aulas

Recursos Pedagógicos: Lousa e giz, caderno, lápis e calculadora.

Metodologia: Será abordado o estudo da Função do Segundo Grau por meio da criptografia RSA. Os alunos deverão escolher dois números primos quaisquer $p<q$, efetuar o produto deles para gerar o valor de n e efetuar a subtração de q por p para gerar o valor de g, logo $n=pq$ e $g=q-p$. Considere a função $f: R\to R$, tal que $f(m)=(m-n-1)^2-4n$. Os alunos serão orientados a substituir o resultado de g^2 no lugar de $f(m)$ e resolver a equação em função de m . Sendo assim, $g^2=(m-n-1)^2-4n$ ou $m^2-2mpq-2m+p^2q^2+1-q^2-p^2=0$. As raízes dessa equação são $m_1=1-(p+q)+pq$ e $m_2=1+p+q+pq$. O importante para o método RSA é apenas a menor raiz, sendo ela o m_1 e $m_1=1-(p+q)+pq=(p-1)\cdot(q-1)$.

Os valores encontrados nas raízes da equação do segundo grau serão discutidos com os alunos, investigando por parte deles uma relação da menor raiz com o valor de n e será mostrado a importância dessa menor raiz no método RSA. Após os alunos terem o domínio do cálculo das raízes de uma Equação do Segundo Grau e perceberem a relação da menor raiz, será explorado os conceitos do vértice da parábola por meio das raízes, ou seja, a média aritmética das raízes resulta no x_v da função, ou seja, $x_v=\frac{1-(p+q)+pq+1+p+q+pq}{2}=\frac{2pq+2}{2}=pq+1$ e também o conceito do y_v, calculado por meio de $f(x_v)$, ou seja, $f(x_v)=(pq+1-pq-1)^2-4pq=-4pq$.

Portanto, nessa função específica será estudado o cálculo das raízes, analisando o padrão que acontece no cálculo do vértice da parábola.

Avaliação: A avaliação será da seguinte maneira: O aluno que encontrou as raízes corretamente e analisou a relação da menor raiz com o valor de n terá a pontuação máxima. O aluno que resolver corretamente todos os exercícios propostos terá a pontuação máxima. Os alunos que não conseguiram resolver ou resolveram parcialmente os exercícios, serão auxiliados para que possam compreender todo o conteúdo ministrado.

5.3 CRIPTOGRAFIA COM MATRIZES

Tema: Criptografia com Matrizes

Objetivo: Essa atividade propõe o estudo da multiplicação e o cálculo da matriz inversa, utilizando-se da criptografia pelas Cifras de Hill.

Conteúdos Relacionados: Produto de Matrizes e Cálculo da Matriz Inversa.

Série de Aplicação: 2º ano do Ensino Médio

Duração: 8 aulas

Recursos Pedagógicos: Lousa e giz, caderno e lápis.

Metodologia: Para codificar uma mensagem, utiliza-se uma tabela de números referente ao alfabeto, escrever uma mensagem com esses números da tabela 6 em forma de uma matriz e multiplicar pela esquerda por uma outra matriz, desde que seja possível essa multiplicação, transformando a mensagem original em um código conforme abaixo:

Código B:

15	24	31	38	19	60	45	21	3
8	19	16	25	14	40	27	12	2

Tabela 7: Código para conversão.

Dada a seguinte tabela:

1 – A	2 – B	3 – C	4 – D	5 – E	6 – F	7 – G	8 – H	9 – I
10 – J	11 – K	12 – L	13 – M	14 – N	15 – O	16 – P	17 – Q	18 – R
19 – S	20 – T	21 – U	22 – V	23 – W	24 – Y	25 – X	26 – Z	27 - vazio

Tabela 8: Alfabeto de conversão.

Decodifique o Código da Tabela 6, sabendo que a chave que os codificou é a matriz $A = \begin{vmatrix} 2 & 1 \\ 1 & 1 \end{vmatrix}$.

Considerando a mensagem M, aplicando a matriz chave A pela esquerda,

obtém-se a mensagem codificada B, ou seja, $AM = B$, para decodificar a mensagem B, aplica-se a matriz inversa A^{-1} em ambos os membros, $A^{-1}AM = A^{-1}B$, logo $M = A^{-1}B$. Portanto para decodificar a mensagem basta calcular a matriz inversa de A e multiplicar pela esquerda o código recebido.

Neste caso $A^{-1} = \begin{vmatrix} 1 & -1 \\ -1 & 2 \end{vmatrix}$, multiplicando por A^{-1} a mensagem B:

$\begin{vmatrix} 1 & -1 \\ -1 & 2 \end{vmatrix} \cdot \begin{vmatrix} 15 & 24 & 31 & 38 & 19 & 60 & 45 & 21 & 3 \\ 8 & 19 & 16 & 25 & 14 & 40 & 27 & 12 & 2 \end{vmatrix}$, obtém-se:

$\begin{vmatrix} 7 & 5 & 15 & 13 & 5 & 20 & 18 & 9 & 1 \\ 1 & 14 & 1 & 12 & 9 & 20 & 9 & 3 & 1 \end{vmatrix}$ que corresponde a mensagem:

$\begin{vmatrix} G & E & O & M & E & T & R & I & A \\ A & N & A & L & I & T & I & C & A \end{vmatrix}$

Após todos conseguirem decodificar a mensagem proposta, os alunos deverão entregar alguma mensagem codificando e decodificando.

Avaliação: A avaliação será da seguinte maneira: O aluno que calculou corretamente a matriz inversa e decodificou corretamente a mensagem terá pontuação máxima. O aluno que não conseguiu calcular a matriz inversa ou que multiplicou a matriz inversa incorretamente no momento de decodificação, será auxiliado por meio de explicação no quadro e proposto um novo exercício de decodificação.

5.4 CRIPTOGRAFIA COM FUNÇÃO DO PRIMEIRO GRAU

Tema: Criptografia com Função do Primeiro Grau

Objetivo: Essa atividade propõe o estudo da função do primeiro grau e da função inversa.

Conteúdos Relacionados: Função do Primeiro Grau e Função Inversa.

Série de Aplicação: 1º ano do Ensino Médio

Duração: 4 aulas

Recursos Pedagógicos: Lousa e giz, caderno e lápis.

Metodologia: Será abordado os conceitos de função do primeiro grau, escolhendo a função $f : R \to R$, tal que $f(x) = 2x + 1$, para codificar a mensagem "MESTRE", o primeiro passo é relacionar a mensagem com os números na Tabela 2: Tabela de conversão, que consta na seção 4.3. Logo a palavra "MESTRE" corresponde aos números: 22 14 28 29 27 14, aplicando cada um dos pontos na função $f(x) = 2x + 1$, tem-se:

$$f(22) = 2 \cdot 22 + 1 = 45$$

$$f(14) = 2 \cdot 14 + 1 = 29$$

$$f(28) = 2 \cdot 28 + 1 = 57$$

$$f(29) = 2 \cdot 29 + 1 = 59$$

$$f(27) = 2 \cdot 27 + 1 = 55$$

$$f(14) = 2 \cdot 14 + 1 = 29$$

Sendo assim, a mensagem codificada é 45 29 57 59 55 29.

Para decodificar a mensagem, precisa-se encontrar a função inversa de $f(x) = 2x + 1$, logo: $f^{-1}(x) = \frac{x-1}{2}$. Para decodificar utiliza-se $f^{-1}(x) = \frac{x-1}{2}$, logo:

$$f^{-1}(45) = \frac{45-1}{2} = 22$$

$$f^{-1}(29) = \frac{29-1}{2} = 14$$

$$f^{-1}(57) = \frac{57-1}{2} = 28$$

$$f^{-1}(59) = \frac{59-1}{2} = 29$$

$$f^{-1}(55) = \frac{55-1}{2} = 27$$

$$f^{-1}(29) = \frac{29-1}{2} = 14$$

Sendo assim, a mensagem decodificada é 22 14 28 29 27 14, que corresponde a palavra "MESTRE".

Avaliação: A avaliação será da seguinte maneira: O aluno que calculou

corretamente a função inversa e decodificou corretamente a mensagem terá pontuação máxima. O aluno que não conseguiu calcular a função inversa ou que calculou incorretamente no momento de decodificação, será auxiliado por meio de explicação no quadro e proposto um novo exercício de decodificação.

5.5 CRIPTOGRAFIA COM FUNÇÃO DO SEGUNDO GRAU

Tema: Criptografia com Função do Segundo Grau

Objetivo: Essa atividade propõe o estudo da função do segundo grau e da função inversa.

Conteúdos Relacionados: Função do Segundo Grau e Função Inversa.

Série de Aplicação: 1º ano do Ensino Médio

Duração: 4 aulas

Recursos Pedagógicos: Lousa e giz, caderno e lápis.

Metodologia: Será abordado os conceitos de função do segundo grau, escolhendo a função $f : R \to R$, tal que $f(x) = x^2 + 1$, para codificar a mensagem "MESTRE", o primeiro passo é relacionar a mensagem com os números na Tabela 2: Tabela de conversão, que consta na seção 4.3. Logo a palavra "MESTRE" corresponde aos números: 22 14 28 29 27 14, aplicando cada um dos pontos na função $f(x) = x^2 + 1$, tem-se:

$f(22) = 22^2 + 1 = 485$

$f(14) = 14^2 + 1 = 197$

$f(28) = 28^2 + 1 = 785$

$f(29) = 29^2 + 1 = 842$

$f(27) = 27^2 + 1 = 730$

$f(14) = 14^2 + 1 = 197$

Sendo assim, a mensagem codificada é 485 197 785 842 730 197.

Para decodificar a mensagem, precisa-se encontrar a função inversa de $f(x) = x^2 + 1$, logo:

$f^{-1}(x) = \sqrt{x-1}$. Para decodificar utiliza-se $f^{-1}(x) = \sqrt{x-1}$, logo:

$f^{-1}(485) = \sqrt{485-1} = 22$

$f^{-1}(197) = \sqrt{197-1} = 14$

$f^{-1}(785) = \sqrt{785-1} = 28$

$f^{-1}(842) = \sqrt{842-1} = 29$

$f^{-1}(730) = \sqrt{730-1} = 27$

$f^{-1}(197) = \sqrt{197-1} = 14$

Sendo assim, a mensagem decodificada é 22 14 28 29 27 14, que corresponde a palavra “MESTRE”.

Avaliação: A avaliação será da seguinte maneira: O aluno que calculou corretamente a função inversa e decodificou corretamente a mensagem terá pontuação máxima. O aluno que não conseguiu calcular a função inversa ou que calculou incorretamente no momento de decodificação, será auxiliado por meio de explicação no quadro e proposto um novo exercício de decodificação.

5.6 CRIPTOGRAFIA COM FUNÇÃO EXPONENCIAL

Tema: Criptografia com Função Exponencial

Objetivo: Essa atividade propõe o estudo da Função Exponencial e a sua Função Inversa.

Conteúdos Relacionados: Função Exponencial e Função Logarítmica.

Série de Aplicação: 1º ano do Ensino Médio

Duração: 6 aulas

Recursos Pedagógicos: Lousa e giz, caderno e lápis.

Metodologia: Será abordado os conceitos de função exponencial, escolhendo a função $f:\ R \to R$, tal que $f(x) = 2^x$, para codificar a mensagem "MESTRE", o primeiro passo é relacionar a mensagem com os números na Tabela 2: Tabela de conversão, que consta na seção 4.3. Logo a palavra "MESTRE" corresponde aos números: 22 14 28 29 27 14, quebrando em bloco de apenas um algarismo para facilitar as contas, obtendo: 2 2 1 4 2 8 2 9 2 7 1 4 e aplicando cada um dos pontos na função $f(x) = 2^x$, tem-se:

$f(2) = 2^2 = 4$

$f(2) = 2^2 = 4$

$f(1) = 2^1 = 2$

$f(4) = 2^4 = 16$

$f(2) = 2^2 = 4$

$f(8) = 2^8 = 256$

$f(2) = 2^2 = 4$

$f(9) = 2^9 = 512$

$f(2) = 2^2 = 4$

$f(7) = 2^7 = 128$

$f(1) = 2^1 = 2$

$f(4) = 2^4 = 16$

Sendo assim, a mensagem codificada é 4 4 2 16 4 256 4 512 4 128 2 16.

Para decodificar a mensagem, precisa-se encontrar a função inversa de $f(x) = 2^x$, logo:

$f^{-1}(x) = \dfrac{\log(x)}{\log(2)}$. Para decodificar utiliza-se $f^{-1}(x) = \dfrac{\log(x)}{\log(2)}$, logo:

$f^{-1}(4) = \dfrac{\log(4)}{\log(2)} = 2$

$f^{-1}(4) = \dfrac{\log(4)}{\log(2)} = 2$

$f^{-1}(2) = \dfrac{\log(2)}{\log(2)} = 1$

$$f^{-1}(16) = \frac{\log(16)}{\log(2)} = 4$$

$$f^{-1}(4) = \frac{\log(4)}{\log(2)} = 2$$

$$f^{-1}(256) = \frac{\log(256)}{\log(2)} = 8$$

$$f^{-1}(4) = \frac{\log(4)}{\log(2)} = 2$$

$$f^{-1}(512) = \frac{\log(512)}{\log(2)} = 9$$

$$f^{-1}(4) = \frac{\log(4)}{\log(2)} = 2$$

$$f^{-1}(128) = \frac{\log(128)}{\log(2)} = 7$$

$$f^{-1}(2) = \frac{\log(2)}{\log(2)} = 1$$

$$f^{-1}(16) = \frac{\log(16)}{\log(2)} = 4$$

Sendo assim, a mensagem decodificada é 2 2 1 4 2 8 2 9 2 7 1 4, reunindo cada bloco com dois algarismo tem-se: 22 14 28 29 27 14 que corresponde a palavra “MESTRE”.

Avaliação: A avaliação será da seguinte maneira: O aluno que calculou corretamente a função inversa e decodificou corretamente a mensagem terá pontuação máxima. O aluno que não conseguiu calcular a função inversa ou que calculou incorretamente no momento de decodificação, será auxiliado por meio de explicação no quadro e será proposto um novo exercício de decodificação.

CONSIDERAÇÕES FINAIS

Inquestionavelmente, a criptografia é um assunto muito interessante que possibilita a segurança na transmissão de informações importantes. O presente trabalho teve como objetivo estudar o método RSA e também articular o tema criptografia com os conteúdos da Matemática. Foram abordadas algumas fórmulas que geram números primos, assim como alguns algoritmos de fatoração, destacando o algoritmo de Fermat, que potencialmente fatora o produto de dois números primos em tempo polinomial, se esses dois números forem próximos.

Foram citados três casos que podem ser estudados na tentativa de quebrar o método RSA (fatorar *n*, encontrar $\varphi(n)$ ou encontrar *d* sem fatorar *n* ou encontrar $\varphi(n)$) e também proposto um método para tentar encontrar $\varphi(n)$. No apêndice encontra-se o algoritmo implementado no Maple12, que utiliza a função $f(m)=(m-n-1)^2-4n$, até gerar o quadrado perfeito, encontrando assim $\varphi(n)=(p-1)\cdot(q-1)$ e por meio da resolução de uma equação do segundo grau, encontram-se os fatores primos p e q.

Por meio da função $f(m)=(m-n-1)^2-4n$, foi possível encontrar os fatores primos com um menor número de repetições em relação ao algoritmo de Fermat, pois pelo método de Fermat, o incremento de x é de apenas uma unidade e calcula-se $y=\sqrt{x^2-n}$, até y gerar um número inteiro, enquanto na função $f(m)=(m-n-1)^2-4n$, subtrai-se quatro unidades e aplica-se o novo ponto até gerar um quadrado perfeito.

RECOMENDAÇÕES PARA ESTUDOS FUTUROS

Para a continuidade deste estudo sobre a criptografia RSA, recomenda-se a análise da função $f(m)=(m-n-1)^2-4n$, tentando delimitar um intervalo mais eficiente de busca do quadrado perfeito, pois no algoritmo que se encontra no apêndice, o critério utilizado para iniciar, foi pela parte inteira da menor raiz da função, avaliando se é múltiplo de quatro, sendo assim, a busca inicia na proximidade do quadrado perfeito 1, ou seja, para números com muitos algarismos, o quadrado perfeito na função será um número muito alto, visto que esse quadrado perfeito é a diferença dos números primos elevado ao quadrado, os quais não são próximos. Assim, sugere-se um estudo para delimitar melhor esse intervalo de busca ou algum método que identifique rapidamente os quadrados perfeitos na função.

REFERÊNCIAS BIBLIOGRÁFICAS

COUTINHO, S.C. **Números Inteiros e Criptografia RSA**. Rio de Janeiro, IMPA, 2011. 226 páginas (Coleção Matemática e Aplicações).

DANTE, L. R. **Matemática: Contexto e aplicações**. São Paulo: Ática , 2010.

EUCLIDES. **Os elementos**. UNESP, 2009. Tradução brasileira por Irineu Bicudo.

HEFEZ, A. **Curso de Álgebra**, vol. 1, Coleção Matemática Universitária, IMPA, Rio de Janeiro, 2002.

HEFEZ, A. Elementos de Aritmética. 2. ed. Rio de Janeiro: SBM, 2011. 176p.

IEZZI, G. [et al.]. **Matemática: ciência e aplicações, 2 : ensino médio**. 6. ed. São Paulo: Saraiva, 2010.

(OLGIN, C.A.;GROENWALD, C.L.O. **Engenharia Didática: Uma experiência com o tema criptografia**. Em: <http://periodicos.uniban.br/index.php/JIEEM/article/viewArticle/214> Acesso em: 27 de setembro de 2015.)

RIBENBOIM, P. **Números Primos. Velhos Mistérios e novos recordes.** 1 ed. Rio de Janeiro: IMPA,2012. 328 p.

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

ROUSSEAU, C.; AUBIN.Y.S. **Matemática e Atualidade.** *Rio de Janeiro: SBM, 2015. 256 páginas (Coleção PROFMAT).*

VOLOCH, J.F. A distribuição dos números primos. Matemática Universitária, número 06, 71-82.

APÊNDICE

ALGORITMO PROGRAMADO NO MAPLE

```
> #N:= p*q;    # M:= (p-1)*(q-1);

> p:= ; q:= ; N:=p*q;
> Digits:= 600;


> f:= x -> x^2 + (M - N - 1)*x + N;

> Delta:= (M - N - 1)^2 - 4*N;

> fDelta:= unapply(Delta,M);

> expand(fDelta(M));

> Raizes_de_M:= solve(Delta,M);

> M1:=floor(min(evalf(Raizes_de_M)))+1;    M2:=floor(max(evalf(Raizes_de_M)));

> resid:=5;
passo4ou1:=1;
while passo4ou1 = 1 do
     M1:= M1-passo4ou1;
     if irem(M1,4) = 0 then
          passo4ou1:=4;
     end if:
end do;

> M1;fDelta(M1);sqrt(fDelta(M1));

> TempoIni:=time();
resid:=5;
while resid > 0 do
     M1:=M1-4;
     resid:= frac(evalf(sqrt(fDelta(M1))))
end do:
TempoFinal:=time()-TempoIni;

> M1;
```