UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA APLICADA

Métodos de Fatoração de Números Inteiros

por

Cristiane Medina Antunes

Dissertação submetida como requisito parcial
para a obtenção do grau de

Mestre em Matemática Aplicada

Prof. Dr. Vilmar Trevisan
Orientador

Porto Alegre, Outubro de 2002.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Antunes, Cristiane Medina

Métodos de Fatoração de Números Inteiros / Cristiane Medina Antunes.—Porto Alegre: PPGMAp da UFRGS, 2002.

75 p.: il.

Dissertação (mestrado) —Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Matemática Aplicada, Porto Alegre, 2002.
Orientador: Trevisan, Vilmar

Dissertação: Matemática Aplicada
Modelo, Dissertação

Métodos de Fatoração de Números Inteiros

por

Cristiane Medina Antunes

Dissertação submetida ao Programa de Pós-Graduação em Matemática Aplicada do Instituto de Matemática da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de

Mestre em Matemática Aplicada

Linha de Pesquisa: Algoritmos Numéricos e Algébricos

Orientador: Prof. Dr. Vilmar Trevisan

Banca examinadora:

Profa. Dra. Cydara Cavedon Ripoll
PPGMAT/UFRGS

Prof. Dr. Claus Haetinger
UNIVATES

Prof. Dr. Rudinei Dias da Cunha
PPGMAp/IM/UFRGS

Dissertação apresentada e aprovada em
31 de Outubro de 2002.

Prof. Dr. Vilmar Trevisan
Coordenador

SUMÁRIO

LISTA DE NOTAÇÕES

$a\|b$	a divide b
$a \nmid b$	a não divide b
$mdc(a,b)$	máximo divisor comum entre a e b
$\lfloor x \rfloor$	maior inteiro menor ou igual a x
$\lceil x \rceil$	menor inteiro maior ou igual a x
$a \equiv b \pmod{m}$	a congruente a b módulo m
$a \not\equiv b \pmod{m}$	a não é congruente a b módulo m
$\left(\frac{a}{p}\right)$	símbolo de Legendre
F_n	número de Fermat
$\|G\|$	ordem do grupo G
$E(a,b)$	grupo de pontos racionais na curva E
$E(a,b)/p$	grupo elíptico módulo p
$\mathbb{Z}/p\mathbb{Z}$	anel dos inteiros módulo p
$\mathcal{O}$	ponto no infinito
$\mathbb{F}_p$	corpo com número finito q de elementos
$N(f)$ ou $Ker(f)$	núcleo do homomorfismo $f : G \to J$

RESUMO

A fatoração de números inteiros é um assunto que, embora muito antigo, desperta cada vez mais interesse. Existem vários métodos de criptografia de chave pública, baseados não só em fatoração de inteiros, mas também em resolução de logaritmos discretos, por exemplo, cuja segurança depende da ineficiência dos métodos de fatoração conhecidos.

Este trabalho tem como objetivo descrever os principais métodos de fatoração utilizados hoje em dia.

Primeiramente, três métodos elementares serão estudados: o método de Fermat e os métodos Rho e $p-1$ de Pollard. A seguir, os dois mais poderosos métodos de fatoração para inteiros sem forma especial: o método de curvas elípticas, e o método de peneira quadrática, os quais tomam como base os métodos $p-1$ e de Fermat, respectivamente.

ABSTRACT

Integer factorization is a subject that, although very old, has drawn more and more interest. There are several public key cryptography methods, based not only on integer factorization, but also on discrete logarithms resolution, for example, whose security depends on the inefficiency of factorization methods.

The main goal of this work is to describe the leading methods of factorization used nowadays.

First, three elementary methods will be studied: Fermat's Method and the methods Rho and $p-1$ of Pollard. Next, the two most powerful factorization methods for integers without special form: Elliptic Curves Method and Quadratic Sieve Method, whose idea are based upon the methods $p-1$ and Fermat, respectively.

1 INTRODUÇÃO

A questão de divisibilidade é, provavelmente, um dos mais antigos problemas de Matemática. Civilizações tão antigas quanto a dos egípcios de dez mil anos atrás, ou a dos maias da América Central, já se preocupavam com esta questão. Os gregos antigos perceberam que todo inteiro poderia ser escrito unicamente como um produto de números primos, e tal descoberta é a base do *Teorema Fundamental da Aritmética* (Teorema 1.1.1).

É surpreendente que um assunto tão antigo possa ao mesmo tempo ser tão atual. A maioria dos métodos de criptografia moderna se baseia no fato de que a fatoração de inteiros é um problema difícil. É fácil construir números compostos grandes multiplicando grandes números primos, no entanto é muito difícil encontrar os fatores primos de um inteiro muito grande. Um exemplo de um sistema que usa este fato é o criptossistema RSA ([5], [7]), o qual é amplamente utilizado hoje em dia.

Para aplicar o RSA, é preciso que se transforme a mensagem composta de palavras em uma seqüência de números. Para isto, constrói-se uma tabela de conversão onde a cada letra é atribuído um número, por exemplo,

A	B	C	$\ldots$	Z
10	11	12		35

O espaço entre duas palavras pode ser representado pelo número 99.

A seguir, escolhem-se dois primos distintos p e q, calcula-se o produto $n = p \cdot q$, e quebra-se a seqüência de números obtida em blocos, os quais devem ser números inteiros menores que n.

Agora o RSA já pode ser aplicado.

A partir dos parâmetros p e q escolhidos anteriormente, além de $n = p \cdot q$, calcula-se

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

Escolhe-se um inteiro positivo e inversível módulo $\phi(n)$, ou seja, tal que $mdc(e, \phi(n)) = 1$. O par (n, e) recebe o nome de *chave de codificação* do criptossistema RSA.

Seja b um inteiro positivo menor que n que representa um bloco de números. O bloco codificado, denotado por $C(b)$, é calculado por

$$C(b) \equiv b^e \ (mod\ n).$$

Cada bloco deve ser codificado separadamente, para que seja possível decodificar a mensagem.

A *chave de decodificação* (n, d), onde $n = p \cdot q$, e d é o inverso de e módulo $\phi(n)$, é usada para decodificar a mensagem. A decodificação também é feita por blocos. Seja a um bloco de mensagem codificada. A decodificação de a, denotada por $D(a)$, é calculada por

$$D(a) \equiv a^d \ (mod\ n).$$

Ao decodificar um bloco codificado, deseja-se obter o bloco correspondente da mensagem original, digamos b, de forma que $D(C(b)) = b$.

Na verdade, não é necessário provar que $D(C(b)) = b$, somente é preciso provar que $D(C(b)) \equiv b \ (mod\ n)$, já que tanto b quanto $D(C(b))$ estão entre 1 e $n-1$, e só serão congruentes módulo n se forem iguais.

Por definição, sabe-se que $D(C(b)) \equiv (b^e)^d \equiv b^{ed} \ (mod\ n)$. Sabe-se também que d é o inverso de e módulo $\phi(n)$, logo

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

de forma que

$$b^{ed} \equiv b \cdot (b^{p-1})^{k(q-1)} \ (mod\ n), \text{ para algum } k \in \mathbb{Z}.$$

Se p não divide b, então $b^{p-1} \equiv 1 \ (mod\ p)$ por Fermat, e $b^{ed} \equiv b \ (mod\ p)$. No caso de p dividir b, $b \equiv 0 \ (mod\ p)$, então $b^{ed} \equiv b \ (mod\ p)$. Portanto, $b^{ed} \equiv b \ (mod\ p)$ para qualquer valor de b.

De forma análoga, pode ser verificado que $b^{ed} \equiv b \ (mod\ q)$, para qualquer valor de q. Portanto, $b^{ed} - b$ é divisível por p e também por q. Como p e q são primos distintos, $mdc(p, q) = 1$, de forma que $pq|(b^{ed} - b)$. Como $n = pq$, pode-se concluir que $b^{ed} \equiv b \ (mod\ n)$, para qualquer inteiro b, o que mostra que o método sempre funciona.

A chave de codificação (n, e) é pública, mas a de decodificação não é. A segurança do RSA depende da dificuldade de descobrir o número d, o qual, junto com n, forma a chave de decodificação. Para calcular d, deve-se aplicar o Algoritmo de Euclides Estendido (Teorema 1.3.3) a e e $\phi(n)$, já que d é o inverso de e módulo $\phi(n)$. Mas para isto, deve-se conhecer $\phi(n)$, ou seja, precisa-se de p e q. Logo, se n for fatorado, descobre-se facilmente a chave de decodificação e quebra-se a mensagem.

Embora o RSA seja o método criptográfico mais empregado atualmente, ele não é o único. Outros sistemas de criptografia bastante usados, como por exemplo aqueles baseados em logaritmos discretos, também se tornam inseguros caso a fatoração de inteiros seja feita eficientemente.

Este trabalho tem como objetivo apresentar alguns dos principais métodos de fatoração de números inteiros utilizados atualmente, e está organizado da seguinte forma.

No capítulo 2 serão apresentados alguns métodos elementares de fatoração: o método de Fermat, e os métodos Rho e $p - 1$ de Pollard. O método de Fermat não é muito utilizado atualmente, mas é a base do método de peneiras quadráticas (capítulo 4). O método $p-1$ serve de modelo para o método das curvas elípticas, que será estudado no capítulo 3, onde primeiramente serão dadas algumas noções de curvas elípticas.

O capítulo 4 abrange o método de peneiras quadráticas (usado para fatorar números em geral) e dá uma breve apresentação do método de peneiras em extensões algébricas dos racionais (usado para fatorar números inteiros da forma $r^e - s$, onde r é um inteiro positivo pequeno, e inteiro positivo e s, um inteiro não nulo de valor absoluto pequeno).

O Teorema Fundamental da Aritmética será enunciado e provado neste primeiro capítulo, onde também serão estudados o algoritmo de fatoração por tentativas e a peneira de Eratóstenes. A peneira de Eratóstenes, que encontra números primos menores que um certo inteiro dado como limite, é usada para acelerar o algoritmo de fatoração por tentativas.

1.1 Teorema Fundamental da Aritmética

O Teorema Fundamental da Aritmética afirma que todo inteiro pode ser escrito como produto de potências de primos e que só há uma escolha possível de primos e expoentes para a fatoração de um inteiro. Há, portanto, dois pontos a serem provados: que a fatoração *existe* e que é *única*.

Teorema 1.1.1 (Teorema Fundamental da Aritmética). *Todo inteiro positivo $n \geq 2$ pode ser representado de modo único (a menos da ordem) como um produto de fatores primos, ou seja,*

$$n = p_1^{e_1} \dots p_h^{e_h},$$

onde $h \geq 1$, $p_1 < p_2 < p_3 < \ldots < p_h$ são números primos e $e_1, \ldots, e_h$ são inteiros positivos.

1.1.1 Existência da Fatoração

A existência da fatoração de um número inteiro n pode ser provada por indução em n.

Se n é primo, acabou. Se n é composto, toma-se $n = 2$ como base de indução. Sabe-se que $2 = 2^1$ (caso em que $h = 1$), o que significa que 2 pode ser fatorado. Supõe-se, por hipótese, que todo número $n \leq k$ pode ser fatorado em um produto de primos.

Agora deve-se verificar se $n = k + 1$ pode ser fatorado em um produto de primos. Primeiro, se $k + 1$ é primo a prova está concluída. No entanto, se $k + 1$ é composto, supõe-se $k + 1 = a \cdot b$, onde $1 < a, b < k$. Mas como $a, b < k$, então pela hipótese a e b podem ser escritos como um produto de primos, digamos,

$$a = p_1 \dots p_r, \text{ e } b = q_1 \dots q_s,$$

de forma que

$$k + 1 = a \cdot b = p_1 \dots p_r q_1 \dots q_s,$$

ou seja, $k+1$ pode ser fatorado em um produto de fatores primos não necessariamente distintos. Assim, foi provado que existe uma fatoração em primos para qualquer número inteiro $n \geq 2$.

1.1.2 Unicidade da Fatoração

Para provar que a fatoração de um inteiro em primos é única, precisa-se de uma propriedade fundamental dos números primos. Para verificá-la começa-se com um lema, que é a primeira aplicação do teorema decorrente do Algoritmo de Euclides Estendido (ver Teorema 1.3.3).

Lema 1.1.1. *Sejam a, b e c inteiros positivos tais que a e b são relativamente primos.*

(1) Se $b|ac$ então $b|c$.

(2) Se $a|c$ e $b|c$, então $ab|c$.

Prova:

(1) Sejam a e b inteiros relativamente primos, isto é, tal que $mdc(a,b) = 1$. Pelo Algoritmo de Euclides Estendido existem inteiros m e n tais que

$$m \cdot a + n \cdot b = 1.$$

Multiplicando esta equação por c, obtém-se

$$m \cdot ac + n \cdot bc = c. \tag{1.1}$$

É óbvio $b|nbc$. Mas por hipótese $b|ac$. Logo $b|(m \cdot ac + n \cdot bc) = c$. Portanto, $b|c$.

(2) Pode ser provado a partir de (1). De fato, se $a|c$, pode-se escrever $c = at$, para algum inteiro t. Mas, $b|c$, e como $mdc(a,b) = 1$, segue da afirmação (1) que $b|t$. Assim, $t = bk$ para algum inteiro k. Portanto,

$$c = at = a(bk) = (ab)k$$

é divisível por ab, o que prova a afirmação (2).

As duas partes deste lema são usadas na demonstração de uma propriedade muito importante dos números primos.

Propriedade Fundamental dos Primos: Seja p um número primo e a e b inteiros positivos. Se $p|ab$, então $p|a$ ou $p|b$.

Prova: Se $p|a$, a prova está pronta. Supõe-se, então, que $p \nmid a$. Como p é primo, então $mdc(a,p) = 1$. Pelo Lema 1.1.1, item (1) segue que $p|b$.

Prova da Unicidade da Fatoração

A prova da unicidade da fatoração de um inteiro na forma do teorema será dada através de redução ao absurdo. Supõe-se que existe algum inteiro que admite mais de uma fatoração na forma estabelecida pelo teorema, e chega-se a uma contradição. Seja, por hipótese, n o menor inteiro positivo que tem duas fatorações

distintas. Pode-se escrever

$$n = p_1^{e_1} \dots p_h^{e_h} = q_1^{r_1} \dots q_s^{r_s} \tag{1.2}$$

onde $p_1 < p_2 < p_3 < \ldots < p_h$ e $q_1 < q_2 < q_3 < \ldots < q_s$ são primos distintos e $e_1, \ldots, e_h, r_1, \ldots, r_s$ são inteiros positivos. Mas por suposição estas fatorações são diferentes.

De acordo com a fatoração da esquerda, p_1 é um primo que divide n. Mas $n = q_1^{r_1} \dots q_s^{r_s}$, segundo a fatoração da direita. A *propriedade fundamental dos primos* garante então que p_1 deve dividir um dos fatores do produto à direita, ou seja, p_1 divide um dos primos da fatoração à direita. Mas um primo só pode dividir outro se forem iguais. Logo p_1 tem que ser um dos primos $q_1,\ q_2\ \ldots$ ou q_h. Digamos que $p_1 = q_j$, onde $1 \leq j \leq s$.

Com isso, pode-se reescrever (1.2) substituindo q_j por p_1

$$\begin{aligned} n = p_1^{e_1} \dots p_h^{e_h} &= q_1^{r_1} \dots q_s^{r_s}. \\ &= q_1^{r_1} \dots p_1^{r_j} \dots q_s^{r_s}. \end{aligned} \tag{1.3}$$

Como há p_1 em ambos os lados da igualdade com multiplicidade ≥ 1, pode-se cancelá-lo obtendo

$$m = p_1^{e_1 - 1} \dots p_h^{e_h} = q_1^{r_1} \dots p_1^{r_j - 1} \dots q_s^{r_s}, \tag{1.4}$$

que é um novo número, menor que n. Mas m tem duas fatorações, que estão escritas em (1.4). Observa-se que estas fatorações foram obtidas a partir das fatorações em (1.2) - que são distintas por hipótese - cancelando-se um termo comum - que é p_1. Logo as fatorações de m em (1.4) são necessariamente distintas.

Tem-se assim um número m menor que n com duas fatorações diferentes. Mas isto não é possível, já que foi assumido que n era o menor inteiro positivo com duas fatorações distintas. Isto significa que a hipótese de que existem duas fatorações distintas leva a um absurdo, confirmando que a fatoração é única.

A seguir, será apresentado um algoritmo bastante simples para determinar os fatores primos e respectivos expoentes de um inteiro n.

1.2 Método de Fatoração por Tentativas

Tendo n como entrada, testa-se sua divisibilidade por cada um dos inteiros de 2 a $n-1$. Começando em 2, itera-se até $n-1$, no caso de n primo, ou até que o quociente seja inteiro, para n composto. Afirmamos que o menor fator encontrado desta maneira tem que ser primo.

De fato, seja f um inteiro tal que $2 \leq f \leq n-1$. Supõe-se que f é o *menor* fator de n e que f' é um fator de f. Portanto, f' também é fator de n. Pela minimalidade de f segue que $f = f'$. Ou seja, o único fator de f maior que 1 é o próprio f. Logo, f é primo.

Antes da descrição detalhada do algoritmo, há ainda uma outra observação a ser feita. O algoritmo consiste em fazer uma busca, começando em 2 e indo até $n-1$, para achar um fator de n. Na verdade, não é necessário procurar fatores maiores que $\lfloor\sqrt{n}\rfloor$. Como o algoritmo que está sendo descrito determina o menor fator de n maior que 1, basta verificar que este fator de n é sempre menor ou igual a $\lfloor\sqrt{n}\rfloor$. Entretanto, há um caso em que isto é evidentemente falso. Se n é primo, então seu menor fator maior que 1 é o próprio n. Portanto, o que tem que ser verificado é que se n é composto e se $f > 1$ é seu menor fator, então $f \leq lfloorsqrtn\rfloor$.

Seja portanto n um número composto e $f > 1$ seu menor fator. Então existe um inteiro positivo a tal que $n = fa$, com $f \leq a$. Mas $a = n/f$, logo $f \leq n/f$. Disto segue que $f^2 \leq n$, que é equivalente a $f \leq \lfloor\sqrt{n}\rfloor$. Resumindo: o algoritmo deve buscar um número que divida n, começando de 2 e avançando até $\lfloor\sqrt{n}\rfloor$. Se n for composto, será encontrado o menor fator de n através deste método, e este fator é *necessariamente primo*. Se nenhum dos números pesquisados é fator de n, isto significa que n é primo.

Assim, dado um inteiro $n > 0$, tem-se uma maneira de determinar se n é primo e, se não for, achar um fator de n. Para achar todos os seus fatores

primos e respectivas multiplicidades basta aplicar o algoritmo várias vezes. Mais precisamente: aplica-se o algoritmo a n encontrando-se o fator q_1, que é o menor fator primo de n. A seguir aplica-se o algoritmo ao quociente n/q_1, determinando um segundo fator primo de n, que será chamado de q_2. É claro que $q_1 \leq q_2$, mas pode acontecer que sejam iguais, basta que q_1^2 divida n. Continuando, aplica-se o algoritmo ao quociente $n/(q_1 q_2)$ e assim por diante. Com isso determina-se uma seqüência crescente de números primos

$$q_1 \leq q_2 \leq q_3 \leq \ldots \leq q_s$$

cada um dos quais é fator de n. A esta seqüência corresponde uma outra, formada pelos quocientes

$$n > \frac{n}{q_1} > \frac{n}{q_1 q_2} > \frac{n}{q_1 q_2 q_3} > \ldots > 0.$$

Observa-se que esta última é uma seqüência *estritamente decrescente* de números inteiros positivos. Como há apenas n inteiros entre n e 1, o algoritmo tem que parar depois de no máximo n laços. Na verdade, é fácil verificar que o menor elemento da seqüência de quocientes (também chamados co-fatores) é sempre 1.

Exemplo: Seja $n = 4500$. Aplicando o algoritmo, descobre-se que o menor fator de 4500 é 2. Aplica-se o algoritmo novamente, só que desta vez ao co-fator de 2 em 4500, que é 4500/2=2250. Com isto descobre-se que 2 também é fator deste número (e portanto de 4500). Com mais cinco execuções deste algoritmo encontra-se que 3^2 e 5^3 são fatores de 4500. Logo, 4500 está completamente fatorado, ou seja, $4500 = 2^2 \cdot 3^2 \cdot 5^3$.

1.2.1 Eficiência do Algoritmo de Fatoração por Tentativas

O algoritmo descrito acima é fácil de entender e programar, mas é ineficaz. No pior caso, ele terá que executar o maior número de laços. Isto ocorre quando n é primo, pois $\lfloor \sqrt{n} \rfloor$ passos terão que ser executados. Como exemplo, considera-se um número primo n de 100 ou mais algarismos. Isto é, $n \geq 10^{100}$ e, portanto, $\lfloor \sqrt{n} \rfloor \geq 10^{50}$. Assim, no mínimo 10^{50} laços terão que ser executados para

que se conclua que n é primo através do algoritmo de fatoração. Se, por exemplo, um computador executa 10^{10} divisões por segundo, com estes dados, tem-se que o computador vai precisar de no mínimo $\frac{10^{50}}{10^{14}} = 10^{40}$ segundos para determinar que n é primo, o que corresponde a 10^{31} anos. Ou seja, é impossível confirmar que um número de 100 ou mais algarismos é primo usando este algoritmo.

Isto não significa que o algoritmo é inútil, pois ao fatorar um inteiro, sempre há a possibilidade que ele tenha um fator primo pequeno, digamos menor que 10^6. O algoritmo acima encontrará tal fator rapidamente.

Este algoritmo pode ser melhorado se for gerada uma seqüência de primos menores ou iguais a $\lfloor\sqrt{n}\rfloor$ através da peneira de Eratóstenes. Então, o algoritmo será aplicado apenas a esses primos e não mais a todos os inteiros $\leq \lfloor\sqrt{n}\rfloor$.

1.3 Peneira de Eratóstenes

Este algoritmo é atribuído ao matemático grego Eratóstenes (276-194 a.C.), que viveu em Cyrene e ensinou em Alexandria. Para encontrar todos os números primos $\leq n$, faz-se uma lista com todos os inteiros de 2 até n. Riscam-se todos os múltiplos de 2 maiores que ele. O primeiro inteiro depois do 2 (ou seja, 3) que não foi cortado deve ser primo. Cortam-se todos os múltiplos de 3 que são maiores que 3. Continua-se desta forma. Ao encontrar um novo primo, cortam-se todos os seus múltiplos que são maiores que o próprio primo e então passa-se para o próximo inteiro que não foi cortado e que deve novamente ser um primo.

Não é preciso chegar a n. Uma vez que foi encontrado um primo maior que a raiz quadrada de n, todos os inteiros restantes que não foram cortados devem ser primos. Se algum deles fosse composto, então ele teria, necessariamente, um fator menor ou igual à sua raiz quadrada. Se $n = a \cdot b$, então $a \leq \lfloor\sqrt{n}\rfloor$ ou $b \leq \lfloor\sqrt{n}\rfloor$.

Este algoritmo tem um sério obstáculo. Se n for muito grande, ele requererá muita memória. Para provar que n é primo, ele tomará aproximadamente $\lfloor\sqrt{n}\rfloor$ ciclos.

A seguir, serão apresentados alguns conceitos básicos que serão usados no decorrer da dissertação (para mais detalhes, ver, por exemplo, [5],[9] ou [18]).

Conceitos Básicos

Definição 1.3.1. *Sejam a e b dois inteiros, a ou b diferente de 0. O* máximo divisor comum *de a e b, denotado por $mdc(a,b)$, é o maior inteiro que divide a e b.*

Definição 1.3.2. *Dados dois inteiros a e b tais que $mdc(a,b) = 1$, diz-se que a e b são* relativamente primos.

Definição 1.3.3. *A função* maior inteiro, *denotada por $\lfloor x \rfloor$, é aquela que associa a cada número real positivo x o maior inteiro menor ou igual a x.*

Teorema 1.3.1 (Algoritmo da Divisão). *Se a e b são inteiros tais que $b > 0$, então existe um único par de inteiros q e r tais que $a = bq + r$, com $0 \leq r < b$.*

Teorema 1.3.2 (Algoritmo de Euclides). *Sejam $r_0 = a$ e $r_1 = b$ inteiros tais que $a \geq b > 0$. Se o algoritmo da divisão é sucessivamente aplicado para obter $r_j = r_{j+1}q_{j+1} + r_{j+2}$ com $0 \leq r_{j+2} < r_{j+1}$ para $j = 0, 1, 2, \ldots, n-2$ e $r_{n+1} = 0$, então $mdc(a,b) = r_n$, o último resto não nulo.*

Teorema 1.3.3 (Algoritmo de Euclides Estendido). *Sejam a e b inteiros positivos. Então*

$$mdc(a,b) = s_n a + t_n b,$$

para $n = 0, 1, 2, \ldots$, onde s_n e t_n são os n-ésimos termos das seqüências definidas recursivamente por

$$s_0 = 1,\ t_0 = 0,$$

$$s_1 = 0,\ t_1 = 1,$$

e

$$s_j = s_{j-2} - q_{j-1}s_{j-1},\ t_j = t_{j-2} - q_{j-1}t_{j-1}$$

para $j = 2, 3, \ldots, n$, *onde os* q_j*'s são os quocientes nas divisões do algoritmo de Euclides quando ele é usado para encontrar* $mdc(a, b)$.

As provas dos três teoremas, 1.3.1, 1.3.2 e 1.3.3, podem ser encontradas em [5] e [17].

Definição 1.3.4. *Sejam* a, b, m *números inteiros, com* $m > 0$. *Diz-se que* a *é congruente a* b *módulo* m, *denotado por* $a \equiv b \pmod{m}$, *se* $m|(a-b)$, *ou seja, se existe inteiro* k *tal que* $a = b + km$.

Definição 1.3.5. *Sejam* a, r, m *inteiros tais que* $m > 0$. *Se* $a \equiv r \pmod{m}$, *então diz-se que* r *é um* resíduo *de* a *módulo* m.

Definição 1.3.6. *Sejam* a *e* m *inteiros com* $mdc(a, m) = 1$. *Diz-se que* a *é um* resíduo quadrático *módulo* m *se a congruência*

$$x^2 \equiv a \pmod{m}$$

tem solução. Caso não tenha solução, então diz-se que a *não é um resíduo quadrático módulo* m.

Uma maneira de identificar se um inteiro é um resíduo quadrático módulo p ou não, é através do símbolo de Legendre. Sejam a um inteiro e $p > 2$ um primo. O *símbolo de Legendre* $\left(\frac{a}{p}\right)$ é definido como segue:

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{se } a \text{ é um resíduo quadrático } \pmod{p}; \\ -1, & \text{se } a \text{ não é resíduo } \pmod{p}; \end{cases}$$

Teorema 1.3.4 (Critério de Euler). *Se* p *for um primo ímpar e* a *um inteiro não divisível por* p, *então*

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}.$$

A prova do critério de Euler pode ser vista em ([7], pag. 43) ou em ([17], pag. 333).

Definição 1.3.7. *Um* número de Fermat *é um número da forma*

$$F_n = 2^{2^n} + 1.$$

Teorema 1.3.5 (Pequeno Teorema de Fermat). *Seja p um número primo, e seja a qualquer número com $a \not\equiv 0 \pmod{p}$. Então*

$$a^{p-1} \equiv 1 \pmod{p}.$$

Definição 1.3.8. *Um conjunto não vazio G com uma operação binária (lei de composição interna) bem definida, digamos $*$, tal que*

- *a operação $*$ é* fechada,
- *a operação $*$ é* associativa,
- *G contém um elemento neutro para $*$,*
- *cada elemento de G tem um inverso em G, com respeito a $*$,*

é chamado grupo.

A ordem de um grupo $(G, *)$, denotada por $|G|$, é o número de elementos de G.

Dizemos que um grupo $(G, *)$ é *abeliano* ou *comutativo* se, e somente se, a operação $*$ é comutativa, isto é,

$$a * b = b * a \quad \forall\ a, b \in G.$$

Dados dois grupos $(G, *)$ e (J, Δ), dizemos que uma aplicação $f : G \to J$ é um *homomorfismo* de G em J se, e somente se,

$$(\forall\ a, b \in G)(f(a * b) = f(a)\Delta f(b)).$$

Sejam $(G, *)$ e (J, Δ) grupos e $f : G \to J$ um homomorfismo. Chama-se *núcleo de* f e denota-se por $N(f)$ ou $Ker(f)$ o seguinte subconjunto de G

$$N(f) = \{x \in G \mid f(x) = u\},$$

onde u indica o elemento neutro de J para $Delta$.

Definição 1.3.9. *Seja A um conjunto não vazio, com duas operações binárias bem definidas: $+$, a qual é denominada adição, e $*$, a qual é denominada multiplicação. Diz-se que $(A, +, *)$ é um* anel *se:*

- $(A, +)$ *é um grupo abeliano,*
- *a multiplicação é associativa,*
- *a multiplicação é distributiva em relação à adição.*

Um anel $(A, +, *)$ é um *anel comutativo* se a multiplicação definida em A é comutativa.

Um *anel com unidade* é um anel $(A, +, *)$ que conta com um elemento neutro para a multiplicação.

Definição 1.3.10. *Um anel K, comutativo com unidade, recebe o nome de* corpo *se todo elemento não nulo de K admite inverso multiplicativo.*

Definição 1.3.11. *Chama-se* característica *de um anel comutativo A, com elemento unidade $u \neq 0$, ao único número natural m tal que $m\mathbb{Z}$ seja o núcleo do homomorfismo $f : \mathbb{Z} \to A$ definido por $f(n) = nu$.*

Por exemplo, para um corpo K com identidade multiplicativa 1, consideram-se os números $2 = 1 + 1$, $3 = 1 + 1 + 1$, $4 = 1 + 1 + 1 + 1$, etc. Se todos estes números são diferentes, diz-se que K tem característica 0. Mas se dois deles forem iguais, então para algum número p tem-se $\underbrace{1 + 1 + 1 + \ldots + 1}_{p \text{ vezes}} = 0$. Se p for escolhido

tão pequeno quanto possível, então prova-se que p é um primo e diz-se que K tem característica p.

Outros conceitos sobre a teoria de grupos, anéis e corpos que se tornem necessários podem ser encontrados em ([19],[20]).

Depois de estudar o método das tentativas, que é fácil de entender e implementar, mas ineficiente na fatoração de números não muito pequenos, passa-se ao estudo de três métodos mais eficientes, mas ainda elementares: o método de Fermat e os métodos Rho e $p-1$ de Pollard.

2 MÉTODOS ELEMENTARES

O algoritmo para divisão por tentativas, apresentado anteriormente, é eficiente apenas quando n é divisível por um primo pequeno. Neste capítulo serão estudados o método de Fermat, e os métodos Rho e $p-1$ de Pollard. O algoritmo inventado por Fermat é muito eficiente quando n tem um fator primo próximo de $\sqrt{n}$. Os métodos de Pollard são chamados algoritmos probabilísticos, e são usados para encontrar fatores pequenos de inteiros grandes.

2.1 Fatoração Através do Método de Fermat

Supõe-se que n é ímpar, pois se n é par, então 2 é um de seus fatores. A idéia é tentar encontrar números inteiros positivos x e y tais que $n = x^2 - y^2$. Supondo que estes números foram encontrados, tem-se

$$n = x^2 - y^2 = (x - y)(x + y).$$

Logo, $(x - y)$ e $(x + y)$ são fatores de n.

Para poder aplicar o algoritmo de Fermat assume-se que há um algoritmo para determinar a raiz quadrada de n. Na verdade, é suficiente obter a parte inteira de $\sqrt{n}$. Se $r \in \mathbb{R}$, sua parte inteira será denotada por $\lfloor r \rfloor$.

O caso mais simples do algoritmo de Fermat ocorre quando n é um quadrado perfeito; isto é, quando existe algum inteiro r tal que $n = r^2$. Neste caso, r é fator de n. Além disso, na notação acima, $x = r$ e $y = 0$. Observa-se que se $y > 0$ então

$$x = \sqrt{n + y^2} > \sqrt{n}.$$

Exemplo: Seja $n = 1342127$ o número a ser fatorado. A variável x é inicializada com a parte inteira da raiz quadrada de n, que neste caso vale $x = 1159$.

Mas

$$1158^2 = 1340964 < 1342127 = n,$$

logo x passa a ser incrementado de um em um. Isto é feito até que

$$\sqrt{x^2 - n}$$

seja inteiro, quando então tomamos este valor para y, ou até que x seja igual a $(n+1)/2$, que neste caso vale 671064. Colocando isto em uma tabela, obtém-se:

Tabela 1

x	$\sqrt{x^2 - n}$
1159	33,97
1160	58,93
1161	76,11
1162	90,09
1163	102,18
1164	113

Assim, um inteiro foi obtido no sexto laço. Portanto, $x = 1164$ e $y = 113$ são os valores desejados. Os fatores correspondentes são $x + y = 1277$ e $x - y = 1051$.

2.1.1 Demonstração do Algoritmo de Fermat

Mas, por que o algoritmo de Fermat funciona, isto é, por que ele pára? Pode-se observar que é necessário considerar separadamente o que acontece quando n é composto, e quando é primo. No primeiro caso, mostra-se que existe um inteiro $x > \lfloor \sqrt{n} \rfloor$ tal que $\sqrt{x^2 - n}$ é um inteiro menor que $(n+1)/2$. Isto significa que se n é composto, então o algoritmo pára antes de chegar a $(n+1)/2$. Se n é primo, então verifica-se que o único valor de x possível é $(n+1)/2$.

Supondo que n pode ser fatorado na forma $n = ab$ onde $a \leq b$, desejam-se obter inteiros positivos x e y tais que $n = x^2 - y^2$. Em outras palavras,

$$n = ab = (x - y)(x + y) = x^2 - y^2.$$

Como $x - y \leq x + y$, tomam-se $a = x - y$ e $b = x + y$. Resolvendo este sistema de duas incógnitas, obtém-se

$$x = \frac{a + b}{2} \text{ e } y = \frac{b - a}{2}.$$

De fato, expandindo os produtos notáveis verifica-se facilmente que

$$\left(\frac{b + a}{2}\right)^2 - \left(\frac{b - a}{2}\right)^2 = ab = n. \tag{2.1}$$

Nota-se que x e y têm que ser números inteiros, mas $(b+a)/2$ e $(b-a)/2$ estão escritos na forma de fração. Porém n é ímpar, por hipótese. Logo, a e b, que são fatores de n, têm que ser ímpares. Portanto, $b + a$ e $b - a$ são pares, e conseqüentemente, $(b + a)/2$ e $(b - a)/2$ são inteiros.

Como definido anteriormente, $n = a \cdot b$, onde $a = x - y$, $b = x + y$ e $a \leq b$. Se n é primo então $n = 1 \cdot n$, ou seja, só se pode ter $a = 1$ e $b = n$. Com isto, $x = (n + 1)/2$; e este é o único valor possível para x. Considera-se agora o caso em que n é composto. Se $a = b$, o algoritmo obtém a resposta desejada já na primeira etapa, caso em que n é um quadrado perfeito. Pode-se, então, supor que n é composto e não é um quadrado perfeito; isto é, que $1 < a < b < n$. Neste caso, afirma-se que o algoritmo vai parar se forem satisfeitas as desigualdades

$$\lfloor \sqrt{n} \rfloor \leq \frac{a + b}{2} < \frac{n + 1}{2}. \tag{2.2}$$

De fato, se n não é um quadrado perfeito, então $\frac{a+b}{2} \geq \lfloor \sqrt{n} \rfloor$. Daí:

$$\begin{aligned} \tfrac{a+b}{2} < \tfrac{n+1}{2} \quad &\Leftrightarrow \quad a + b < n + 1 = ab + 1 \\ &\Leftrightarrow \quad a + b - b - 1 < ab + 1 - b - 1 \\ &\Leftrightarrow \quad a + 1 < ab - b \\ &\Leftrightarrow \quad 1 < b. \end{aligned}$$

Este argumento mostra que $1 < b$ é equivalente à desigualdade original. Como $1 < a < b$ vale por hipótese, está provado que $(a+b)/2 < (n+1)/2$.

Considera-se agora a desigualdade da esquerda. Observa-se primeiro que, como $\lfloor\sqrt{n}\rfloor \leq \sqrt{n}$, basta verificar que $n \leq (a+b)^2/4$ é verdadeira. Mas, por (2.1), tem-se que

$$\frac{(b+a)^2}{4} - n = \frac{(b-a)^2}{4},$$

que é sempre um número não negativo. Assim, foi obtido $(a+b)^2/4 - n \geq 0$, que é equivalente à desigualdade desejada.

Voltando ao algoritmo, a variável x é inicializada com o valor $\lfloor\sqrt{n}\rfloor$ e vai sendo incrementada de uma unidade a cada laço. Assim, (2.2) garante que, se n for composto, $(a+b)/2$ será alcançado antes que se chegue a $(n+1)/2$. Quando $x = (a+b)/2$,

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

pela identidade (2.1). Atingindo este laço, o algoritmo pára, obtendo a e b como fatores. Portanto, se n é composto, o algoritmo sempre pára antes de chegar a $x = (n+1)/2$, tendo determinado os fatores de n.

Este método tem algumas características importantes. A principal delas é que os laços não envolvem multiplicações ou divisões, de forma que eles executam extremamente rápido. O problema é o enorme número de execuções requeridas.

O método de Fermat funciona na direção oposta à da divisão por tentativas. Neste último, começa-se procurando fatores pequenos e continua-se até a raiz quadrada de n. No método de Fermat começa-se procurando fatores próximos à raiz quadrada de n e continua-se procurado fatores decrescentes.

Atualmente, este algoritmo não é muito implementado a menos que se saiba que o número a ser fatorado tem dois fatores que estão relativamente próximos à sua raiz quadrada. Mas ele contém a idéia por trás de um dos mais poderosos algoritmos usados hoje para fatorar números com grandes fatores primos, o método da Peneira Quadrática (capítulo 4).

2.2 Métodos de Pollard

Em 1974 e 1975, John Pollard anunciou dois novos algoritmos para encontrar fatores de inteiros grandes. Cada algoritmo executa uma seqüência de operações polinomiais (adições, subtrações e multiplicações) de tal forma que os resultados intermediários são compostos e não nulos.

Os métodos de Pollard são chamados algoritmos probabilísticos, pois começa a ser introduzido acaso nos procedimentos. Agora, não se tem mais certeza de encontrar um fator de um dado tamanho em uma quantidade fixada de tempo. Mas, em compensação, usualmente um fator será encontrado em muito menos tempo do que através de um algoritmo determinístico.

2.2.1 Método Rho

Este método foi inicialmente chamado **Método de Monte Carlo** devido à sua natureza pseudo-randômica. Agora é mais popularmente conhecido como **Método Rho**.

Supõe-se que n é um inteiro grande, composto, e que p é seu menor divisor primo. O objetivo é escolher inteiros $x_0, x_1, \ldots, x_s$ de forma que estes inteiros tenham resíduos não negativos mínimos distintos, módulo n, mas seus resíduos não negativos mínimos módulo p não sejam todos distintos. Como se pode ver, usando argumentos probabilísticos (ver [9] ou [14], por exemplo), é provável que este seja o caso quando s é grande comparado a $\sqrt{p}$ mas pequeno quando comparado a $\sqrt{n}$, e os números são escolhidos randomicamente.

Uma vez que tenham sido encontrados inteiros x_i e x_j onde $0 \leq i < j \leq s$ tais que $x_i \equiv x_j \pmod{p}$ mas $x_i \not\equiv x_j \pmod{n}$, segue que $mdc(x_i - x_j, n)$ é um divisor não trivial de n, já que $x_i - x_j$ é divisível por p, mas não por n. O número $mdc(x_i - x_j, n)$ pode ser encontrado rapidamente usando-se o algoritmo de Euclides. Entretanto, encontrar $mdc(x_i - x_j, n)$ para cada par (i, j)

com $0 \leq i < j \leq s$ requer que sejam encontrados $O(s^2)$ máximos divisores comuns ([3]). A seguir, mostra-se, inicialmente, como calcular os x_i, e logo depois, como reduzir o número de vezes em que o algoritmo de Euclides precisa usado.

Para encontrar tais inteiros x_i e x_j, começa-se com um valor inicial x_0, que é escolhido randomicamente, e uma função polinomial $f(x)$ arbitrária com coeficientes inteiros e grau maior que 1. Calculam-se os termos x_k, $k = 1, 2, 3, \ldots$, usando a definição recursiva

$$x_{k+1} \equiv f(x_k) \pmod{n},\ 0 \leq x_{k+1} < n.$$

O polinômio $f(x)$ deve ter a propriedade que a seqüência $x_0, x_1, \ldots, x_k, \ldots$ se comporta como uma seqüência verdadeiramente aleatória.[1] O exemplo a seguir ilustra como esta seqüência é gerada.

Exemplo 1: Seja $n = 8051$ e suponha que $x_0 = 2$ e $f(x) = x^2 + 1$. Encontra-se $x_1 = 5$, $x_2 = 26$, $x_3 = 677$, $x_4 = 7474$, $x_5 = 2839$, $x_6 = 871$, e assim por diante.

Nota-se, pela definição recursiva de x_k, que se

$$x_i \equiv x_j \pmod{p},$$

onde p é um inteiro positivo, então

$$x_{i+1} \equiv f(x_i) \equiv f(x_j) \equiv x_{j+1} \pmod{p}.$$

Segue que, se $x_i \equiv x_j \pmod{p}$, então a seqüência x_k se torna periódica módulo p com um período $j - i$. Isto é, $x_q \equiv x_r \pmod{p}$ sempre que $q \equiv r \pmod{j-i}$, e $q \geq i$ e $r \geq i$. Disto, pode-se ver que se s é o menor múltiplo de $j - i$ que é, no mínimo, igual a i, então $x_s \equiv x_{2s} \pmod{p}$.

Para encontrar um fator de n, deve-se encontrar o mdc de $x_{2k} - x_k$ e n para $k = 1, 2, 3, \ldots$. Sabe-se que tal fator de n foi encontrado quando aparecer

[1] É claro que não é uma seqüência aleatória, pois foi usada uma regra para gerar seus termos. Entretanto, esta seqüência deve ter termos com as mesmas propriedades de uma seqüência aleatória.

um valor k para o qual $1 < x_{2k} - x_k < n$. Das observações, pode-se perceber que é provável que tal k seja encontrado próximo a $\sqrt{p}$.

Na prática, o polinômio $f(x) = x^2 + 1$ freqüentemente é escolhido para gerar a seqüência de inteiros $x_0, x_1, x_2, \ldots, x_k, \ldots$, pois um polinômio linear $f(x) = ax + b$ não será suficientemente randômico para este propósito. Além disso, o valor inicial $x_0 = 2$ também é usado freqüentemente. Esta escolha de valor inicial e de polinômio produz uma seqüência que se comporta como uma seqüência randômica quando este método de fatoração é aplicado.

Exemplo 2: Encontrar um fator não trivial de $n = 8051$ usando o método Rho com valor inicial $x_0 = 2$ e polinômio gerador $f(x) = x^2 + 1$.

No exemplo anterior encontrou-se $x_1 = 5$, $x_2 = 26$, $x_3 = 677$, $x_4 = 7474$, $x_5 = 2839$, $x_6 = 871$. Usando o algoritmo de Euclides segue que $mdc(x_2 - x_1, 8051) = mdc(26-5, 8051) = mdc(21, 8051) = 1$ e $mdc(x_4 - x_2, 8051) = mdc(7474-26, 8051) = mdc(7448, 8051) = 1$. Entretanto, um fator não trivial de 8051 é encontrado no próximo passo, pois $mdc(x_6 - x_3, 8051) = mdc(871 - 677, 8051) = mdc(194, 8051) = 97$. Logo 97 é um fator de 8051.

Pode-se ver por que este método é chamado de método Rho observando a figura abaixo, a qual mostra o comportamento da seqüência x_i onde $x_0 = 2$ e $x_{i+1} = x_i^2 + 1 \pmod{97}$, $i \geq 1$. Nota-se que existe uma parte não periódica, que ocorre antes da periodicidade e é a cauda do Rho, e uma parte periódica que é o laço.

O método Rho é prático para a fatoração de inteiros com fatores primos moderadamente grandes.

Em 1980, R.P. Brent propôs uma mudança na forma do algoritmo. A fim de evitar armazenar muitos valores dos $x_i's$, ele sugeriu o cálculo das diferenças:

$$x_1 - x_3$$

$$x_3 - x_6$$
$$x_3 - x_7$$

$$x_7 - x_{12}$$
$$x_7 - x_{13}$$
$$x_7 - x_{14}$$
$$x_7 - x_{15},$$

e em geral:

$$x^{2^n-1} - x_j,\ 2^{n+1} - 2^{n-1} \leq j \leq 2^{n+1} - 1.$$

O importante é a diferença entre coordenadas, a qual apenas aumenta um a cada vez. A menor coordenada é mantida até garantir que já se saiu da cauda.

Já que muitos mdc's têm que ser calculados, usualmente milhares ou dezenas de milhares, obtém-se uma economia substancial de tempo se for tomado o produto de, digamos, dez valores sucessivos de $(x_i - x_j)$ (mod n) e se depois for calculado o mdc daquele produto com n. Se for verificado que o mdc é n, volta-se àqueles últimos dez valores e faz-se o mdc de n com cada um deles. Na prática, entretanto, se n divide o produto de dez diferenças sucessivas, ele freqüentemente divide exatamente uma destas diferenças.

Em geral, pode-se esperar que o número de ciclos necessários seja em torno da raiz quadrada do menor primo dividindo n. Isto se deve ao fato de que

a seqüência dos x_i usualmente se comporta como se eles fossem aleatórios. A probabilidade de que não haja repetições entre os primeiros t termos da seqüência é então

$$\frac{p-1}{p} \cdot \frac{p-2}{p} \cdot \ldots \cdot \frac{p-(t-1)}{p},$$

e esta probabilidade diminui cerca de 50% quando t se aproxima de $\sqrt{p}$.

2.2.2 Método $p-1$

O Pequeno Teorema de Fermat 1.3.5 é a base deste método de fatoração inventado por J.M. Pollard em 1974, e que ficou conhecido como método $p-1$. Com ele pode-se encontrar um fator não trivial de um inteiro n que tem um fator primo p tal que os primos que dividem $p-1$ são relativamente primos e pequenos ([4],[7],[21]).

Para ver como este método funciona, supõe-se que se deseja encontrar um fator do inteiro positivo n. Além disso, supõe-se que n tem um fator primo p tal que $p-1$ divide $k!$, onde k é um inteiro positivo que não seja grande demais, ou seja, deseja-se que $p-1$ tenha somente fatores primos pequenos. Por exemplo, se $p = 2269$, então $p-1 = 2268 = 2^2 3^4 7$, de forma que $p-1$ divide $9!$, mas não divide nenhum valor menor da função fatorial.

O motivo pelo qual se quer que $p-1$ divida $k!$ é para que se possa aplicar o pequeno teorema de Fermat (Teorema 1.3.5). Como $p-1$ divide $k!$, então $k! = (p-1)q$ para algum inteiro q. Portanto,

$$2^{k!} = 2^{(p-1)q} = (2^{p-1})^q \equiv 1^q = 1 \ (\mod p)$$

o que implica que p divide $2^{k!}-1$, de forma que $M = (2^{k!}-1) - nt$ para qualquer inteiro t. Logo, p divide M, já que p divide ambos, $2^{k!}-1$ e n.

Agora, para encontrar um divisor de n é preciso somente calcular $d = mdc(M, n)$. Isto pode ser feito rapidamente usando o algoritmo de Euclides. Para este divisor d ser um divisor não trivial, é necessário que M seja não nulo.

Este é o caso quando o próprio n não divide $2^{k!}-1$, o que é provável quando n tem divisores primos grandes.

Para usar este método, deve-se calcular $2^{k!}$ módulo n, para $k = 0,\ 1,\ \ldots$. Isto pode ser feito eficientemente via exponenciação modular. Para encontrar o menor resto positivo de $2^{k!}$ módulo n fixa-se $r_1 = 2^{0!}$ e usa-se a seguinte seqüência de cálculos:

$$r_2 \equiv r_1^2 \ (\mod n),\ r_3 \equiv r_2^3 \ (\mod n),\ \ldots,\ r_k \equiv r_{k-1}^k \ (\mod n),$$

pois $2^{(n+1)!} = 2^{(n+1)n!} = (2^{n!})^{(n+1)}$. Este procedimento é ilustrado no seguinte exemplo:

Exemplo: Para encontrar $2^{9!} \ (\mod 5157437)$ a seguinte seqüência de cálculos é realizada:

$$\begin{aligned}
r_2 &\equiv r_1^2 = 2^2 \equiv 4 \ (\mod 5157437)\\
r_3 &\equiv r_2^3 = 4^3 \equiv 64 \ (\mod 5157437)\\
r_4 &\equiv r_3^4 = 64^4 \equiv 1304905 \ (mod\ 5157437)\\
r_5 &\equiv r_4^5 = 1304905^5 \equiv 404913 \ (\mod 5157437)\\
r_6 &\equiv r_5^6 = 404913^6 \equiv 2157880 \ (\mod 5157437)\\
r_7 &\equiv r_6^7 = 2157880^7 \equiv 4879227 \ (\mod 5157437)\\
r_8 &\equiv r_7^8 = 4879227^8 \equiv 4379778 \ (\mod 5157437)\\
r_9 &\equiv r_8^9 = 4379778^9 \equiv 4381440 \ (\mod 5157437)
\end{aligned}$$

Segue que

$$2^{9!} \equiv 4381440 \ (\mod 5157437).$$

O exemplo a seguir ilustra o uso do método $p-1$ de Pollard para encontrar um fator do inteiro 5157437.

Exemplo: Para fatorar 5157437 usando o método $p-1$ encontra-se sucessivamente r_k, o menor resíduo positivo de $2^{k!}$ módulo 5157437, para $k = 1\ , 2\ , 3\ , \ldots$, como foi feito no exemplo anterior. Calcula-se $mdc(r_k - 1, 5157437)$ em cada passo. Encontrar um fator de 5157437 requer nove passos, porque $mdc(r_k - 1, 5157437) = 1$ para $k = 1,\ 2,\ \ldots,\ 8$, mas $mdc(r_9 - 1,\ 5157437) = mdc(4381439, 5157437) = 2269$. Segue que 2269 é um divisor de 5157437.

Na prática, não se sabe quão próximo de n deve-se chegar antes de encontrar o seu primeiro divisor primo. E não é desejável ir tão longe tal que todos eles sejam encontrados. Por esta razão, periodicamente deve-se checar o valor de $mdc(c^{k!} - 1, n)$, onde c é um inteiro positivo relativamente primo a n. Se ainda for 1, continua-se. Se for n, então já se ultrapassou todos os divisores de n, sendo necessário tentar um valor diferente de c, ou tentar um algoritmo diferente. É claro que se não for nem 1 nem n, então encontrou-se o divisor não trivial procurado.

Este método nem sempre funciona. Entretanto, como o método não depende da base escolhida, pode-se estendê-lo e encontrar fatores de outros inteiros usando bases diferentes de 2, desde que tais bases sejam relativamente primas a n.

Na prática, a primeira tentativa de fatorar um inteiro grande n é fazer divisão por tentativas por todos os primos menores que n, ou então aplicar o método de Fermat, se já se sabe que existem fatores próximos à raiz de n. A seguir, o método Rho ou o $p-1$ são usados para procurar fatores primos de tamanho intermediário (digamos até 10^5, por exemplo). Somente depois destes métodos terem falhado as ferramentas mais poderosas, tais como a peneira quadrática e o método das curvas elípticas, devem ser utilizadas.

O método das curvas elípticas, que será estudado no próximo capítulo, toma o método $p-1$ de Pollard como base.

3 MÉTODO DAS CURVAS ELíPTICAS

O método $p-1$ encontra um fator p de n se $p-1$ é suficientemente suave[1]. Entretanto, ele falha se $p-1$ tem fatores primos grandes. Em 1985, Hendrik Lenstra Jr. ([10]) superou esta dificuldade quando anunciou um método similar chamado **Método das Curvas Elípticas** (MCE), o qual será estudado neste capítulo.

O método $(p-1)$ de Pollard (capítulo 2) é baseado no fato que os elementos não nulos em $\mathbb{Z}/p\mathbb{Z}$ formam um grupo multiplicativo $(\mathbb{Z}/p\mathbb{Z})^*$ de ordem $p-1$; assim, se $p-1|k$, então $a^k = 1$ no grupo. Supõe-se que n tem um fator primo p tal que $p-1$ é um produto de primos pequenos. Pelo Pequeno Teorema de Fermat (Teorema 1.3.5),

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

logo, p divide $mdc(a^{p-1}, n)$. Como não se conhece p, escolhem-se valores aleatórios para k e calcula-se $mdc(a^{k-1}, n)$ até que $1 < mdc(a^{k-1}, n) \leq k$.

A idéia de Lenstra é substituir o grupo multiplicativo pelo grupo de pontos em uma curva elíptica $E(a,b)/p$, e substituir o inteiro a por um ponto $P \in E(a,b)/p$. Como no algoritmo de Pollard, escolhe-se um inteiro k composto de um produto de primos pequenos. Então, se o número de elementos em $E(a,b)/p$ divide k, tem-se $kP = \mathcal{O}$, onde $\mathcal{O}$ é um ponto no infinito e é o elemento neutro em $E(a,b)/p$. E da mesma forma que antes, o fato que $kP = \mathcal{O}$ geralmente permitirá que se encontre um fator não trivial de n.

3.1 Introdução

Uma curva elíptica é uma curva com a forma

$$Ax^3 + Bx^2y + Cxy^2 + Dy^3 + Ex^2 + Fxy + Gy^2 + Hx + Iy + J = 0, \qquad (3.1)$$

[1] Dizemos que um número inteiro é suave se os fatores primos que o compõem são todos pequenos

onde $A,\ B,\ \ldots,\ J$ são elementos de um corpo K, com $A, G \neq 0$, e cuja principal propriedade é que se uma reta a corta em dois pontos, então também a cortará em um terceiro ponto.

Por uma mudança apropriada de variáveis, uma curva elíptica geral 3.1 sobre um corpo de característica $\neq 2, 3$ (definição 1.3.11) pode ser escrita na forma

$$y^2 = ax^3 + bx + c, \tag{3.2}$$

onde $4a^3 + 27b^2 \neq 0$, o que garante que a equação (3.2) não tem fatores repetidos. Se K tem característica três, então o termo x^2 não pode ser eliminado e a curva é transformada em

$$y^2 = x^3 + ax^2 + bx + c. \tag{3.3}$$

Se K tem característica dois, então a situação é ainda pior. Uma forma geral na qual uma curva elíptica sobre qualquer corpo K pode ser transformada é chamada a *forma de Weierstrass* e é dada por

$$y^2 + ay = x^3 + bx^2 + cxy + dx + e, \tag{3.4}$$

onde $a,\ b,\ c,\ d,\ e$ são elementos de K. Felizmente $\mathbb{Q},\ \mathbb{R},\ \mathbb{C}$ têm característica zero, e a equação 3.2 pode ser usada.

Como exemplos, podem-se citar as seguintes curvas elípticas simples:

$$\begin{aligned} E_1 : y^2 &= x^3 + 17, \\ E_2 : y^2 &= x^3 + x, \\ E_3 : y^2 &= x^3 - 4x^2 + 16. \end{aligned}$$

3.2 Soluções Racionais em Curvas Elípticas

É interessante estudar soluções de curvas elípticas, isto é, pares (x, y) que satisfazem (3.3), em números racionais, e em inteiros módulo p. Cada um dos três exemplos tem soluções em inteiros. Por exemplo,
E_1 tem as soluções (-2,3), (-1,4) e (2,5),

E_2 tem a solução (0,0),
E_3 tem as soluções (0,4) e (4,4).

No entanto, neste trabalho, procuram-se apenas soluções racionais. Isto se deve ao fato que os pontos racionais em uma curva elíptica formam um grupo abeliano finitamente gerado ([20]).

Estas soluções foram encontradas por tentativa e erro, ou seja, foram testados valores pequenos de x para ver se $x^3 + ax^2 + bx + c$ é um quadrado perfeito. Da mesma forma, checando uns poucos valores pequenos para x, descobre-se a solução racional (1/4, 33/8) para E_1.

Para criar mais soluções, usa-se a interação entre Geometria e Teoria dos Números. Por exemplo, tomando-se uma reta passando pelo ponto $(-1, 0)$ no círculo unitário, procura-se o outro ponto onde a reta corta o círculo. Prova-se que, tomando-se retas cuja inclinação é um número racional, o segundo ponto de interseção também terá coordenadas (x, y) racionais. Desta forma, usam-se retas passando pelo ponto (-1,0) para criar novos pontos com coordenadas racionais.

O mesmo tipo de método pode ser usado para encontrar pontos com coordenadas racionais em curvas elípticas. No entanto, nem sempre ele nos fornece um tal ponto.

Exemplo: Como ilustração, considera-se E_1

$$y^2 = x^3 + 17.$$

Desenham-se retas através do ponto $P = (-2, 3)$ para ver quais outros pontos serão encontrados. Por exemplo, supondo que a reta com inclinação 1 é escolhida, obtém-se

$$y - 3 = x + 2.$$

Para encontrar a interseção desta reta com E_1, substitui-se $y = x + 5$ na equação para E_1 e resolve-se para x, obtendo-se

$$x^3 - x^2 - 10x - 8 = 0.$$

Como, neste caso, uma das soluções do polinômio cúbico já é conhecida, não é necessário resolver o polinômio. Ambas, a curva elíptica E_1 e a reta, passam através do ponto $P = (-2, 3)$, então $x = -2$ deve ser uma raiz. Isto permite a fatoração do polinômio cúbico como

$$x^3 - x^2 - 10x - 8 = (x + 2)(x^2 - 3x - 4).$$

Agora pode-se usar a fórmula quadrática para encontrar as raízes $x = -1$ e $x = 4$ de $x^2 - 3x - 4$. A substituição destes valores na equação da reta $y = x + 5$ fornece as coordenadas y dos novos pontos (-1,4) e (4,9). Estes pontos satisfazem a equação $y^2 = x^3 + 17$. A seguir, mostra-se que nem sempre isto acontece. Para esta mesma curva, toma-se a reta com inclinação 3, passando pelo ponto $P = (-2, 3)$. Esta reta tem equação

$$y - 3 = 3(x + 2),$$

a qual depois de rearranjada fica

$$y = 3x + 9.$$

Substitui-se $y = 3x + 9$ na equação para E_1, obtendo-se

$$(x + 2)(x^2 - 11x - 32) = 0.$$

Exatamente como antes, pode-se usar a fórmula quadrática para encontrar as raízes de $x^2 - 11x - 32$. No entanto, os dois valores encontrados são

$$x = \frac{11 \pm \sqrt{249}}{2}.$$

Este obviamente não é o tipo de resposta desejada, visto que procuram-se pontos em E_1 que tenham coordenadas racionais.

Para saber por quê isto acontece, desenha-se a reta L de inclinação m através do ponto $P = (-2, 3)$ e encontra-se sua interseção com E_1. A reta L é dada pela equação

$$L : y - 3 = m(x + 2). \tag{3.5}$$

Para encontrar a interseção de L e E_1, substitui-se $y = m(x+2)+3$ na equação para E_1 e resolve-se para x. Quando se faz isso, obtém-se a seguinte equação cúbica para resolver

$$\begin{aligned} y^2 &= x^3 + 17 \\ (m(x+2)+3)^2 &= x^3 + 17 \\ 0 &= x^3 - m^2x^2 - (4m^2+6m)x - (4m^2+12m-8). \end{aligned}$$

Sabe-se que uma das raízes é $x = -2$, então a equação é fatorada como

$$0 = (x+2)(x^2 - (m^2+2)x - (2m^2+6m-4)).$$

Entretanto, as outras duas raízes provavelmente não são números racionais[2]

Resumindo: a idéia de usar retas através de pontos conhecidos para produzir novos pontos tem alguns obstáculos. Como vimos acima, o problema é que se tem um polinômio cúbico com uma raiz racional, e isto deixa as outras duas raízes como soluções de um polinômio quadrático cujas raízes podem não ser racionais. Mas sabe-se que se um polinômio quadrático com coeficientes inteiros ou racionais tem uma raiz racional, então a outra raiz também será racional. Então, fazendo com que um polinômio cúbico com coeficientes inteiros ou racionais original tenha duas raízes racionais, garante-se que a terceira raiz também será racional.

Isto mostra bem o problema. O polinômio cúbico original tem uma raiz racional porque foi escolhida uma reta passando através do ponto $P = (-2, 3)$, dessa forma assegurando que $x = -2$ é uma raiz. Como mencionado anteriormente, a principal característica de uma curva elíptica é que se uma reta não vertical a corta em dois pontos, então também a cortará em um terceiro ponto. Para fazer com que o polinômio cúbico tenha duas raízes racionais, deve ser escolhida uma reta que já passe por dois pontos racionais distintos na curva elíptica E_1.

[2]A equação quadrática $x^2 - (m^2+2)x - (2m^2+6m-4) = 0$ tem soluções racionais se e só se $(m^2+2)^2 + 4(2m^2+6m-4)$ for um quadrado perfeito em $\mathbb{Q}$, o que nem sempre acontece, como vimos no exemplo.

Um exemplo ilustrará esta idéia. Começa-se com os dois pontos $P = (-2, 3)$ e $Q = (2, 5)$ na curva elíptica

$$y^2 = x^3 + 17.$$

A reta que conecta P e Q tem inclinação $(5-3)/(2-(-2)) = 1/2$, então sua equação é

$$y = \frac{1}{2}x + 4.$$

Substituindo isto na equação para E_1 obtém-se

$$\begin{aligned} y^2 &= x^3 + 17 \\ \left(\tfrac{1}{2}x + 4\right)^2 &= x^3 + 17 \\ 0 &= x^3 - \tfrac{1}{4}x^2 - 4x + 1. \end{aligned}$$

$x = -2$ e $x = 2$ devem ser duas das raízes, então a equação acima pode ser fatorada como

$$0 = (x-2)(x+2)\left(x - \frac{1}{4}\right).$$

Nota-se que a terceira raiz é de fato um número racional, $x = 1/4$, e substituindo este valor na equação da reta obtém-se a coordenada y correspondente, $y = 33/8$. Assim, tomando a reta através de duas soluções de coordenadas racionais distintas conhecidas, (-2,3) e (2,5), encontra-se a solução racional (1/4,33/8) para a curva elíptica E_1. O gráfico a seguir mostra a interseção da curva elíptica E_1 com a reta que passa pelos pontos $P = (-2, 3)$ e $Q = (2, 5)$, e que tem inclinação $m = 1/2$.

Se este procedimento for repetido usando a nova solução (1/4,33/8) e se a reta for desenhada através de (-2,3) e (1/4,33/8), digamos, sabe-se qual será o terceiro ponto de interseção com E_1, isto é, (2,5). Sabe-se também que se (x, y) é um ponto numa curva elíptica E_1, então o ponto $(x, -y)$ também será um ponto em E_1. Isto está claro pela simetria de E_1 sobre o eixo $\overrightarrow{OX}$. Então deve-se tomar o novo ponto, (1/4,33/8), substituí-lo por (1/4,-33/8), e então repetir o procedimento acima usando a reta através de (1/4,-33/8) e (-2,3). Esta reta tem inclinação 19/6

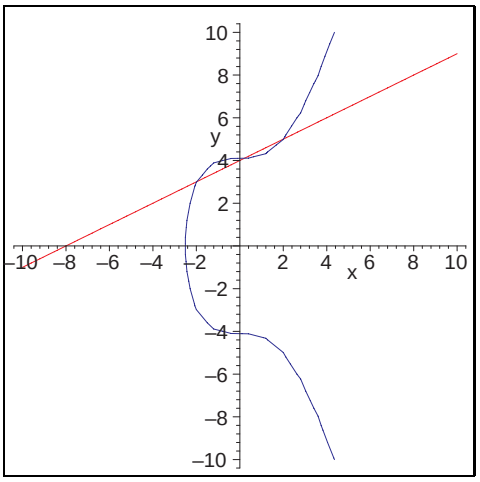

Figura 3.1: Gráfico da interseção entre as curvas $y^2 = x^3 + 17$ e $y = \frac{1}{2}x + 4$.

e é dada pela equação $y = 19x/6 + 28/3$. Substituindo na equação para E_1, deve-se encontrar as raízes de

$$x^3 - \frac{361}{36}x^2 - \frac{190}{9}x + \frac{53}{9} = 0.$$

Como 1/4 e -2 são duas das raízes, então pode-se dividir este polinômio cúbico por $(x - 1/4)(x + 2)$ para encontrar a outra raiz,

$$x^3 - \frac{361}{36}x^2 - \frac{190}{9}x + \frac{53}{9} = \left(x - \frac{1}{4}\right)(x + 2)\left(x - \frac{106}{9}\right).$$

Isto dá $x = 106/9$, e substituindo este valor de x na equação da reta obtém-se $y = -1097/27$. Assim, foi encontrado um novo ponto, (106/9,-1097/27), satisfazendo a equação

$$y^2 = x^3 + 17.$$

O fato de que todas as soluções racionais para E_1 podem ser obtidas a partir de um conjunto gerador finito é um caso especial do famoso teorema, apresentado a seguir.

Teorema 3.2.1 (Teorema de Mordell). *Seja C uma curva elíptica dada pela equação*

$$C : y^2 = x^3 + ax^2 + bx + c,$$

onde a, b, c são inteiros tais que o discriminante

$$\Delta(C) = -4a^3 + a^2b^2 - 4b^3 - 27c^2 + 18abc$$

é não nulo. Então há uma lista finita de soluções

$$P_1 = (x_1, y_1),\ P_2 = (x_2, y_2),\ \ldots,\ P_r = (x_r, y_r),$$

com coordenadas racionais de forma que toda solução racional para C pode ser obtida começando com estes r pontos e repetidamente tomando retas através de pares de pontos, interseccionando com C, e refletindo para criar novos pontos.

Mordell provou este teorema em 1922. Infelizmente, a prova (vide [20]) é complicada demais para ser dada em detalhes e está fora do escopo deste trabalho, mas o seguinte *sketch* da prova de Mordell mostra que ela nada mais é do que uma versão extravagante do método da descida, de Fermat:

1. O primeiro passo é fazer uma lista $P_1, P_2, \ldots, P_r$, de pontos em C que tenham coordenadas racionais.

2. O próximo passo é mostrar que se Q é qualquer ponto com coordenadas racionais que não está na lista, então é possível escolher um dos $P_i's$ de forma que a reta através de P_i e Q corte C em um terceiro ponto Q' o qual é menor que Q.

3. Repetindo este processo, obtém-se uma lista de pontos $Q, Q', Q'', Q''', \ldots$ de tamanho decrescente, e mostra-se que finalmente o tamanho fica tão pequeno que chega-se a um dos $P_i's$ na lista original.

Agora serão analisadas algumas das soluções racionais para E_1. Começa-se com $P_1 = (-2, 3)$ e $P_2 = (-1, 4)$. A reta através de P_1 e P_2 corta E_1 em um terceiro ponto, o qual é refletido sobre o eixo $\overrightarrow{OX}$ e chamado P_3. A seguir toma-se a reta através de P_1 e P_3, cujo ponto de interseção com E_1 é refletido sobre o eixo $\overrightarrow{OX}$ para obter P_4. Usando a reta através de P_1 e P_4, obtém-se similarmente P_5, e assim por diante. A seguinte tabela lista os primeiros $P_n's$. Como se pode ver, o número de dígitos dos numeradores e denominadores cresce com uma rapidez assustadora.

$$P_1 = (-2, 3),\ P_2 = (-1, 4),\ P_3 = (4, -9),\ P_4 = (2, 5),\ P_5 = \left(\frac{1}{4}, \frac{-33}{8}\right),$$

$$P_6 = \left(\frac{106}{9}, \frac{1097}{27}\right),\ P_7 = \left(\frac{-2228}{961}, \frac{-63465}{29791}\right),\ P_8 = \left(\frac{76271}{289}, \frac{-21063928}{4913}\right),$$
$$P_9 = \left(\frac{-9776276}{6145441}, \frac{54874234809}{15234548239}\right),\ P_{10} = \left(\frac{3497742218}{607770409}, \frac{-215890250625095}{14983363893077}\right).$$

Uma forma quantitativa de medir o "tamanho" destes pontos é olhar para o numerador e o denominador das coordenadas x. Em outras palavras, se as coordenadas de P_n como frações irredutíveis,

$$P_n = \left(\frac{A_n}{B_n}, \frac{C_n}{D_n}\right),$$

pode-se definir o tamanho de P_n como

$$\begin{aligned} tamanho(P_n) &= máximo\ de\ |A_n|\ e\ |B_n|. \\ &= max\{A_n, B_n\}. \end{aligned}$$

Por exemplo,

$$tamanho(P_1) = max\{|-2|, |1|\} = 2$$

e

$$tamanho(P_7) = max\{|-2228|, |961|\} = 2228.$$

O número de dígitos em tamanho (P_n) é cn^2. Usando métodos mais avançados, prova-se que c é aproximadamente 0.1974 ([3]). Em outras palavras, para grandes valores de n, o tamanho de P_n deve ser

$$número\ de\ dígitos\ no\ tamanho(P_n) \approx\ 0.1974n^2,$$

$$tamanho(P_n) \approx\ 10^{0.1974n^2}\ \approx\ (1.574)^{n^2}.$$

O método das curvas elípticas se baseia no fato de que os pontos em uma curva elíptica sobre um corpo formam um grupo abeliano quando a operação de grupo é adequadamente definida.

Há uma maneira bem conhecida de definir tal grupo e operação. A definição desta operação repete precisamente a idéia utilizada nos exemplos acima. Como uma tangente à curva é considerada como tendo dois pontos de interseção no

ponto de tangência, pode-se calcular o ponto de interseção extra usando o seguinte lema.

Lema 3.2.1. *Sejam* (x_1, y_1) *e* (x_2, y_2) *dois pontos na curva elíptica dada por*

$$y^2 = x^3 + ax + b, \quad 4a^3 + 27b^2 \neq 0.$$

Assume-se que se $x_1 = x_2$ *então* $y_1 \neq -y_2$. *Permite-se, entretanto, que os dois pontos sejam iguais desde que* $y_1 \neq 0$. *O terceiro ponto de interseção,* (x_3, y_3), *é calculado da seguinte forma:*
Se $x_1 \neq x_2$, *então estabelece-se*

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2}.$$

Se $x_1 = x_2$, *então estabelece-se*

$$\lambda = \frac{3{x_1}^2 + a}{2y_1}.$$

Então tem-se que

$$x_3 = \lambda^2 - x_1 - x_2,$$

$$y_3 = \lambda \cdot (x_3 - x_1) + y_1.$$

Prova: A quantidade λ é a inclinação da reta que conecta os dois pontos. Isto está claro se $x_1 \neq x_2$. Isto precisa ser provado se os dois $x's$ são iguais. Visto que ambos os pontos satisfazem a equação (3.2), tem-se que

$${y_1}^2 - {y_2}^2 = {x_1}^3 - {x_2}^3 + a \cdot (x_1 - x_2),$$

$$(y_1 - y_2) \cdot (y_1 + y_2) = (x_1 - x_2) \cdot ({x_1}^2 + x_1 \cdot x_2 + {x_2}^2 + a),$$

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2} = \frac{{x_1}^2 + x_1 \cdot x_2 + {x_2}^2 + a}{y_1 + y_2}. \tag{3.6}$$

Conforme x_2 se aproxima de x_1, o lado direito da equação tende a

$$\frac{3{x_1}^2 + a}{2y_1}.$$

A equação (3.6) é válida para qualquer par de pontos na reta, então tem-se também que

$$\lambda \cdot (y_3 + y_1) = {x_3}^2 + x_3 \cdot x_1 + {x_1}^2 + a,$$

$$\lambda \cdot (y_3 + y_2) = {x_3}^2 + x_3 \cdot x_2 + {x_2}^2 + a.$$

Subtraindo a segunda equação da primeira, obtém-se

$$\lambda \cdot (y_1 - y_2) = x_3 \cdot (x_1 - x_2) + ({x_1}^2 - {x_2}^2).$$

Dividindo ambos os lados por $x_1 - x_2$, obtém-se

$$\lambda \cdot \lambda = x_3 + x_1 + x_2,$$

do qual pode-se calcular x_3. O cálculo de y_3 segue da definição de derivada da reta

$$\lambda = \frac{y_3 - y_1}{x_3 - x_1},$$

onde, isolando o valor de y_3, obtém-se:

$$y_3 = \lambda(x_3 - x_1) + y_1.$$

Observação: Note-se que a soma de dois pontos em uma reta não é o terceiro ponto naquela reta, mas sim a reflexão daquele terceiro ponto sobre o eixo $\overrightarrow{OX}$, e este novo ponto pertence, pela simetria de 3.2, à mesma curva elíptica.

Corolário 3.2.1. *Sejam os pontos (x_1, y_1) e (x_2, y_2) como definidos acima. Se ambos os pontos têm coordenadas racionais, então o terceiro ponto também terá coordenadas racionais.*

Definição 3.2.1. *Dada uma curva elíptica e dois pontos racionais naquela curva: (x_1, y_1) e $(x_2, y_2) \neq (x_1, -y_1)$, define-se uma operação binária + por*

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3),$$

onde x_3 e y_3 são definidos pelo Lema 3.2.1.

Agora tem-se um conjunto, a saber os pontos racionais em uma curva elíptica, e uma operação binária. Deseja-se transformar isto em um grupo. Para tal, é preciso definir $(x, y) + (x, -y)$, e precisa-se também encontrar uma identidade e inversos. Pode-se resolver todos estes problemas com um único passo.

Definição 3.2.2. *Define-se ∞ como a identidade para a operação binária $+$ e*

$$(x, y) + (x, -y) = (x, -y) + (x, y) = \infty.$$

O ponto ∞ pode ser pensado como um ponto no infinito de forma que toda reta vertical passe por ele. Uma das belezas desta definição é que agora *toda* reta que intercepta a curva em dois pontos também intercepta em um terceiro ponto. O gráfico 3.2 mostra a interseção da curva elíptica E_1 com a reta vertical $x = 2$.

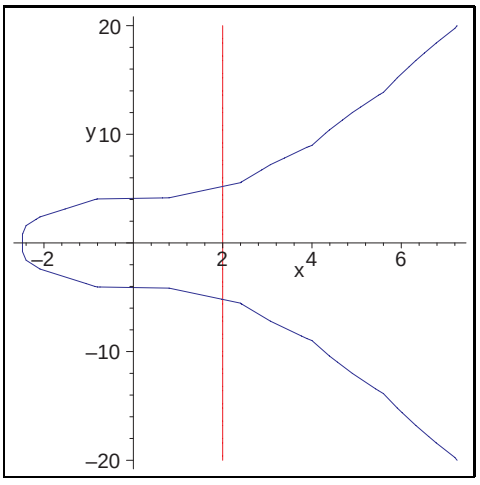

Figura 3.2: Gráfico da interseção entre as curvas $y^2 = x^3 + 17$ e $x = 2$.

Proposição 3.2.1. *Dada uma curva elíptica,*

$$y^2 = x^3 + ax + b, \quad 4a_2 + 27b^2 \neq 0,$$

seja $E(a, b)$ o grupo de pontos racionais na curva, juntamente com o ponto ∞ no infinito com a operação binária como definida acima.

3.3 Curvas Elípticas Módulo p

Definição 3.3.1. *Todas as operações aritméticas entre números racionais fazem perfeitamente sentido módulo n, desde que os denominadores sejam relativamente primos a n. Especificamente, define-se a operação* + módulo n *por*

$$\infty \;=\; \mathcal{O} \;=\; \textit{identidade}.$$

Se $x_1 \equiv x_2 \ (\mod n)$ e $y_1 \equiv -y_2 \ (\mod n)$, então

$$(x_1, y_1) + (x_2, y_2) = \mathcal{O}.$$

Se $x_1 \not\equiv x_2 \ (\mod n)$ e se $mdc(x_1 - x_2, n) = 1$, então seja s o inverso de $x_1 - x_2$ módulo n e define-se λ por

$$\lambda = (y_1 - y_2) \cdot s \ (\mod n).$$

Se $x_1 \equiv x_2 \ (\mod n)$ e se $mdc(y_1 + y_2, n) = 1$, então $y_1 \equiv y_2 \ (\mod n)$, e então seja s o inverso de $2y_1$ módulo n e define-se λ por

$$\lambda = (3 \cdot {x_1}^2 + a) \cdot s \ (\mod n).$$

Define-se x_3 e y_3 por

$$x_3 = (\lambda^2 - x_1 - x_2) \ (\mod n),$$

$$y_3 = (\lambda \cdot (x_3 - x_1) + y_1) \ (\mod n).$$

A operação binária + módulo n é dada por

$$(x_1, y_1) + (x_2, y_2) \equiv (x_3, -y_3) \ (\mod n),$$

onde x_3 e y_3 estão definidos. Em particular, se n é um primo ímpar então a operação binária está sempre bem definida.

Definição 3.3.2. *Seja p um primo maior que 3 e sejam a e b inteiros escolhidos tais que*

$$4a^3 + 27b^2 \not\equiv 0 \ (\mod p).$$

Então $E(a,b)/p$ denota o grupo elíptico módulo p cujos elementos são pares (x,y) de inteiros não negativos menores que p satisfazendo

$$y^2 \equiv x^3 + ax + b \ (\mod p),$$

junto com a identidade, ∞, e cuja operação binária é dada por $+$ módulo p como definido acima.

Dado (x_1, y_1) em $E(a,b)/p$, define-se

$$(x_i, y_i) \equiv (x_1, y_1)^i \ (\mod p).$$

Como um exemplo, seja $p = 5$, $a = b = -1$. Os pontos de $E(-1,-1)/5$ devem satisfazer

$$y^2 \equiv x^3 - x - 1 \ (\mod 5).$$

Note que x não pode ser 3 porque 23 não é um resíduo quadrático módulo 5. Os elementos deste grupo são

$$(0,2),\ (1,2),\ (2,0),\ (4,2),$$

$$(0,3),\ (1,3),\ (4,2),\ \infty.$$

Se $(x_1, y_1) = (0, 2)$, então

$$
\begin{array}{rcll}
(x_2, y_2) &=& (0,2)+(0,2) & \lambda \equiv (3\cdot 0-1)\cdot 4 \equiv 1 \pmod 5,\\
& & & x_2 \equiv 1-0-0 \equiv 1 \pmod 5,\\
& & & -y_2 \equiv 1\cdot(1-0)+2 \equiv 3 \pmod 5,\\
&=& (1,2); & \\
(x_3, y_3) &=& (1,2)+(0,2) & \lambda \equiv (2-2)\cdot 1 \equiv 0 \pmod 5,\\
& & & x_3 \equiv 0-1-0 \equiv 4 \pmod 5,\\
& & & -y_3 \equiv 0\cdot(4-0)+2 \equiv 2 \pmod 5,\\
&=& (4,3); & \\
(x_4, y_4) &=& (4,3)+(0,2) & \lambda \equiv (3-2)\cdot 4 \equiv 4 \pmod 5,\\
& & & x_4 \equiv 16-4-0 \equiv 2 \pmod 5,\\
& & & -y_4 \equiv 4\cdot(2-0)+2 \equiv 0 \pmod 5,\\
&=& (2,0); & \\
(x_5, y_5) &=& (2,0)+(0,2) & \lambda \equiv (0-2)\cdot 3 \equiv 4 \pmod 5,\\
& & & x_5 \equiv 16-2-0 \equiv 4 \pmod 5,\\
& & & -y_5 \equiv 4\cdot(4-0)+2 \equiv 3 \pmod 5,\\
&=& (4,2); & \\
(x_6, y_6) &=& (4,2)+(0,2) & \lambda \equiv (2-2)\cdot 4 \equiv 0 \pmod 5,\\
& & & x_6 \equiv 0-4-0 \equiv 1 \pmod 5,\\
& & & -y_6 \equiv 0\cdot(1-0)+2 \equiv 2 \pmod 5,\\
&=& (1,3); &
\end{array}
$$

$$
\begin{array}{rcll}
(x_7, y_7) &=& (1,3)+(0,2) & \lambda \equiv (3-2)\cdot 1 \equiv 1 \pmod 5,\\
& & & x_7 \equiv 1-1-0 \equiv 0 \pmod 5,\\
& & & -y_7 \equiv 1\cdot(0-0)+2 \equiv 2 \pmod 5,\\
&=& (0,3); & \\
(x_8, y_8) &=& (0,3)+(0,2) \equiv \infty & .
\end{array}
$$

Exatamente como nas técnicas para fatoração e teste de primalidade, a chave para os métodos das curvas elípticas está em conhecer a ordem de $E(a, b)/p$.

Esta ordem pode ser avaliada observando-se que para toda classe de resíduos módulo p, se $x^3 + ax + b$ é um resíduo quadrático não nulo, então há dois valores de y que correspondem àquele x; se $x^3 + ax + b$ é divisível por p, então há um valor de y que corresponde àquele x, a saber, $y = 0$; e caso contrário não há valores de y que correspondem àquele valor de x. Como há também um ponto no infinito, pode-se expressar a ordem de $E(a,b)/p$ em termos do símbolo de Legendre:

$$|E(a,b)/p| = 1 + \sum_{x=1}^{p} \left(\left(\frac{x^3 + ax + b}{p} \right) + 1 \right).$$

Infelizmente, esta fórmula é totalmente impraticável para grandes valores de p. No entanto, sabe-se muito sobre a ordem de $E(a,b)/p$. O teorema a seguir, publicado em 1934, é devido a Helmut Hasse, e diz que esta ordem é da forma $p + 1 - t_p$, onde t_p é um inteiro que depende da curva e de p, para o qual $|t_p| \leq 2\sqrt{p}$.

Teorema 3.3.1. *Seja M o número de $\mathbb{F}_p$-pontos em uma curva elíptica definida sobre $\mathbb{F}_p$. Então*

$$|M - (p+1)| \leq 2\sqrt{p}.$$

Ou seja, as ordens dos grupos de curvas elípticas módulo p estão uniformemente distribuídas sobre o intervalo $I(p) = (p + 1 - 2\sqrt{p},\ p + 1 + 2\sqrt{p})$. A prova deste teorema pode ser vista em ([22]).

3.4 Método das Curvas Elípticas

De muitas formas, as curvas elípticas são análogos naturais de grupos multiplicativos, mas elas têm a vantagem que há muito mais flexibilidade em escolher uma curva elíptica do que em escolher um corpo finito. A idéia de Lenstra é substituir o grupo multiplicativo usado no método $p - 1$ pelo grupo de pontos em uma curva elíptica $E(a,b)/p$, e substituir o inteiro a por um ponto $P \in E(a,b)/p$. Como no algoritmo de Pollard, escolhe-se um inteiro k composto de um produto de

pequenos primos. Então, se o número de elementos em $E(a,b)/p$ divide k, tem-se $kP = \mathcal{O}$, onde $\mathcal{O}$ é o elemento neutro em $E(a,b)/p$, o que geralmente permitirá que um fator não trivial de n seja encontrado.

Se for escolhida somente uma curva C com coeficientes inteiros e forem consideradas suas reduções módulo vários primos, então não há vantagem em se usar o algoritmo de Lenstra. Para uma única curva C, obtém-se sucesso se há algum primo p dividindo n de forma que $|E(a,b)/p|$ é um produto de pequenos primos. Similarmente, obtém-se sucesso usando o algoritmo de Pollard se há um primo p dividindo n de forma que $p-1$ é um produto de primos pequenos. Mas, supondo que há falha, no algoritmo de Pollard se não se obtém sucesso então é o fim. Mas o algoritmo de Lenstra é mais flexível e permite que se continue fazendo outras tentativas. Em outras palavras, pode-se escolher uma nova curva elíptica e começar tudo de novo. Visto que $|E(a,b)/p|$ varia consideravelmente para um primo p fixo e uma curva C variável, as chances de se obter sucesso são bastante boas.

Agora estes comentários vagos são transformados em um algoritmo explícito. Nota-se, na Seção 1, que se C é uma curva cúbica não singular com coeficientes em $E(a,b)/p$, então

$$|E(a,b)/p| = p + 1 - \epsilon_p \; com \; |\epsilon_p| \leq 2\sqrt{p}.$$

Além disso, pode-se mostrar que conforme C varia sobre todas estas curvas, os números ϵ_p estão bem distribuídos sobre o intervalo de $(-2\sqrt{p})$ a $(+2\sqrt{p})$. Então é bastante provável (mas ainda não rigorosamente provado) que nos depararemos muito rapidamente com uma curva C com $|E(a,b)/p|$ igual a um produto de primos pequenos.

Então aqui, formalmente exposto, está o algoritmo de Lenstra.

Algoritmo da Curva Elíptica de Lenstra

Seja $n \geq 2$ um inteiro composto para o qual se procura um fator.

(1) Checar que $mdc(n, 6) = 1$ e que n não tem a forma m^r para algum $r \geq 2$.

(2) Escolher aleatoriamente inteiros $b,\ x_1,\ y_1$ entre 1 e n.

(3) Seja $c \equiv {y_1}^2 - {x_1}^3 - bx_1 \pmod{n}$, e seja C a curva cúbica $C : y^2 = x^3 + bx + c$, e $P = (x_1, y_1) \in C$.

(4) Verificar que $mdc(4b^3 + 27c^2, n) = 1$. (Se for igual a n, voltar e escolher um novo b. Se está estritamente entre 1 e n, então é um fator não trivial de n, e então está pronto.)

(5) Escolher um número k o qual é um produto de primos pequenos com potências pequenas. Por exemplo, toma-se

$$k = mmc(1, 2, 3, \ldots, K)$$

para algum inteiro K.

(6) Calcular

$$kP = \left(\frac{a_k}{d_k^2}, \frac{b_k}{d_k^3}\right).$$

(7) Calcular $D = mdc(d_k, n)$. Se $1 < D < n$, então D é um fator não trivial de n e está acabado. Se $D = 1$, ou volta para (5) e aumenta k ou volta para (2) e escolhe uma nova curva. Se $D = n$, então volta para (5) e diminui k.

Há duas coisas sobre este algoritmo que devem ser discutidas: por que ele funciona, e como executar o passo (6), que parece envolver o cálculo

$$\underbrace{P + P + \ldots + P}_{k\ vezes}.$$

Para ver porque o algoritmo funciona, supõe-se que a curva C e o número k são escolhidos de forma que, para algum primo p que divide n, tem-se $|E(a, b)/p|$ dividindo k. Então todo elemento em $E(a, b)/p$ tem ordem dividindo

k, então, em particular, se o ponto $P \in E(a,b)/p$ for tomado e reduzido módulo p, sabe-se que

$$\tilde{k}P = k\tilde{P} = \tilde{\mathcal{O}}.$$

Em outras palavras, a redução de kP módulo p é o ponto no infinito $\mathcal{O}$, então deve-se ter p dividindo d_k. Portanto, p dividirá $mdc(d_k, n)$. Além disso, é extremamente provável que n não dividirá d_k, o que significa que um fator não trivial de n será encontrado. Isto explica porque o algoritmo de Lenstra funciona.

Uma maneira eficiente de calcular kP, obviamente, não é somando $P + P + \ldots + P$, k-vezes. Em vez disso, usa-se o mesmo método de expansão binária usado para calcular a^k. Primeiro, escreve-se k como

$$k = k_0 + k_1 \cdot 2 + k_2 \cdot 2^2 + k_3 \cdot 2^3 + \ldots + k_r \cdot 2^r$$

com cada k_i igual a 0 ou 1. Pode-se fazer isso com $r \leq log_2 k$. Em seguida, calcula-se

$$\begin{aligned} P_0 &= P \\ P_1 = 2P_0 \quad &= 2P \\ P_2 = 2P_1 \quad &= 2^2 P \\ P_3 = 2P_2 \quad &= 2^3 P \\ \vdots \quad &\quad \vdots \\ P_r = 2P_{r-1} \quad &= 2^r P. \end{aligned}$$

Finalmente, calcula-se

$$kP = (\textit{soma dos } P_i's \textit{ para os quais } k_i = 1).$$

Então pode-se calcular kP em menos que $2log_2 k$ passos de duplicação e soma de pontos.

Nota-se, entretanto, que não é interessante calcular as coordenadas de kP como números racionais porque os numeradores e denominadores teriam aproximadamente k^2 dígitos. Até mesmo para valores relativamente pequenos, tais como $k \approx 10^{50}$, isto leva a números com uma quantidade enorme de dígitos. Então é muito melhor realizar todos os cálculos módulo n.

Mas n não é primo, então como usar as fórmulas para duplicar e somar os pontos? Considera-se o problema de somar dois pontos, digamos $Q_1 = (x_1, y_1)$ e $Q_2 = (x_2, y_2)$, onde x_1, x_2, y_1, y_2 são inteiros módulo n. A fórmula para $Q_3 = Q_1 + Q_2$ diz que

$$x_3 = \lambda^2 - x_1 - x_2, \text{ e } y_3 = \lambda \cdot (x_3 - x_1) + y_1,$$

onde

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2}.$$

A dificuldade está em calcular λ porque o anel $\mathbb{Z}/n\mathbb{Z}$ não é um corpo, então $x_2 - x_1$ pode não ter inverso. Ao tentar fazer este cálculo, encontram-se três possíveis resultados:

(1) $mdc(x_2 - x_1, n) = 1$

Neste caso, $x_2 - x_1$ tem um inverso em $\mathbb{Z}/n\mathbb{Z}$, logo pode-se calcular Q_3 módulo n. (Talvez deva ser mencionado que se $mdc(a, n) = 1$, então o algoritmo de Euclides pode ser modificado para dar uma solução para a equação $ax \equiv 1 \pmod{n}$. Assim, há uma maneira rápida de encontrar o inverso de a módulo n.)

(2) $1 < mdc(x_2 - x_1, n) < n$

Neste caso não se pode encontrar Q_3, mas o inteiro $mdc(x_2 - x_1, n)$ fornece o fator de n desejado. Logo, o algoritmo pode ser terminado aqui.

(3) $mdc(x_2 - x_1, n) = n$

Neste caso, o melhor a fazer é voltar ao passo (5) e reduzir o valor de k; ou então pode-se voltar ao passo (2) e escolher uma nova curva.

Similarmente, para duplicar um ponto $Q = (x, y)$ módulo n, é preciso calcular a razão

$$\lambda = \frac{f'(x)}{2y} = \frac{3x^2 + a}{2y} \pmod{n}.$$

Então se obtém as três alternativas: ou se pode calcular $2Q$ ($\mod n$); ou se obtém um fator não trivial de n; ou $mdc(y, n) = n$ e deve-se recomeçar com um novo k ou uma nova curva.

Isto, em resumo, é como o algoritmo da curva elíptica de Lenstra funciona; embora na prática haja várias maneiras de torná-lo mais eficiente.

Para ilustrar o procedimento geral, será dado um exemplo de fatoração de um inteiro utilizando-se o algoritmo de Lenstra.

3.5 Exemplo

Seja $n = 1715761513$ o inteiro a ser fatorado através do método das curvas elípticas.

O primeiro passo é verificar que n não é primo. Usando o método dos quadrados sucessivos, calcula-se facilmente

$$2^{n-1} \equiv 93082891 \ (\mod n).$$

Pelo Pequeno Teorema de Fermat (1.3.5), isto prova que n não é um primo. Agora já se pode procurar um fator.

O primeiro passo do algoritmo de Lenstra verifica que n não é uma potência perfeita. Usando uma calculadora, calculam-se:

$$\sqrt{n},\ \sqrt[3]{n},\ \sqrt[4]{n},\ \ldots,\ \sqrt[31]{n} \approx 1.9855.$$

Nenhum deles é inteiro, então n não é uma potência perfeita.

Visto que $\sqrt{1715761513} \approx 42422$, sabe-se que n tem algum fator primo p menor que 42422. Deseja-se escolher um valor de k de forma que algum inteiro próximo a p divida k. Tenta-se

$$k = mmc(1, 2, 3, \ldots, 17) = 12252240,$$

o qual tem muitos fatores menores que 42422.

A seguir, precisam-se escolher uma curva elíptica e um ponto naquela curva. Como foi indicado na descrição do algoritmo de Lenstra, é mais fácil determinar o ponto P e um dos coeficientes da curva, e depois escolher o outro coeficiente de forma que o ponto esteja na curva. Assim, estabelece-se o ponto

$$P = (2, 1),$$

tomam-se vários valores de b, e depois se estabelece $c = -7 - 2b$. Para começar, seja $b = 1$, então $c = -9$. Desse modo, encontram-se a curva C e o ponto dado por

$$C : y^2 = x^3 + x - 9,\ P = (2, 1) \in C.$$

O objetivo é calcular $kP \pmod{n}$ usando sucessivas duplicações, então a primeira coisa a fazer é expressar k como uma soma de potências de 2. Isto é facilmente efetuado, fornecendo

$$\begin{aligned} k &= 12252240 = \\ &= 2^4 + 2^6 + 2^{10} + 2^{12} + 2^{13} + 2^{14} + 2^{15} + 2^{17} + 2^{19} + 2^{20} + 2^{21} + 2^{23}. \end{aligned}$$

Logo, a fim de calcular $kP \pmod{n}$, é preciso determinar $2^iP \pmod{n}$ para $0 \leq i \leq 23$. Os resultados são reunidos na tabela abaixo.

Tabela 2	
i	2^iP (mod 1715761513)
0	(2, 1)
1	(1286821173, 1072350709)
2	(1334478523, 112522703)
3	(912789305, 77695868)
4	(385062894, 618628731)
5	(866358838, 450284374)
6	(904716938, 169383608)
7	(808696477, 1201030016)
8	(572301268, 107111567),
9	(1512647092, 1695275444)
10	(1858186, 1224662922)
11	(1550404618, 825515387)
12	(1519325194, 1657497846)
13	(522917322, 524407354)
14	(25207285, 1375034461)
15	(781360494, 1457273929)
16	(1108412304, 25813532)
17	(435914774, 323718902)
18	(1399483199, 1203611423)
19	(778823593, 192206539))
20	(853199887, 1012680972)
21	(501929966, 910060788)
22	(1315182921, 305331854)
23	(257200250, 318342966)

Finalmente, somando os pontos apropriados nesta tabela, encontra-se o valor de kP (mod n):

$$
\begin{aligned}
2^4P &= 16P = (385062894,\ 618628731)\\
(2^4+2^6)P &= 80P = (831572269,\ 1524749605)\\
(2^4+2^6+2^{10})P &= 1104P = (1372980126,\ 736595454)\\
(2^4+2^6+2^{10}+2^{12})P &= 5200P = (1247661424,\ 958124008)\\
(\textit{soma parcial anterior})+2^{13}P &= 13392P = (1548582473,\ 1559853215)\\
(\textit{soma parcial anterior})+2^{14}P &= 29776P = (201510394,\ 7154559)\\
(\textit{soma parcial anterior})+2^{15}P &= 62544P = (629067322,\ 264081696)\\
(\textit{soma parcial anterior})+2^{17}P &= 193616P = (844665131,\ 537510825)\\
(\textit{soma parcial anterior})+2^{19}P &= 717904P = (886345533,\ 342856598)\\
(\textit{soma parcial anterior})+2^{20}P &= 1766480P = (370579416,\ 1254954111)\\
(\textit{soma parcial anterior})+2^{21}P &= 3863632P = (77302130,\ 514483068)\\
(\textit{soma parcial anterior})+2^{23}P &= 12252240P = (1225303014,\ 142796033)
\end{aligned}
$$

Sabe-se agora que na curva $y^2 = x^3 + x - 9$ considerada módulo n, tem-se

$$kP = 12252240(2,1) \equiv (421401044, 664333727) \pmod{1715761513}.$$

Mas isto não diz nada a respeito dos fatores de n. O ponto importante do algoritmo de Lenstra é que ele dá um fator de n precisamente quando a lei de adição falha. Então se realmente se pode calcular $kP \pmod{n}$, então deve-se começar de novo, ou com um novo valor de k, ou com um novo ponto P, ou com uma nova curva.

Escolhe-se a última alternativa e varia-se a curva. Então continua-se com $k = 12252240$ e com $P = (2,1)$, mas agora toma-se $b = 2$ e $c = -7 - 2b = -11$. Usando esta curva e repetindo o cálculo acima, descobre-se que novamente é possível calcular $kP \pmod{n}$. Então agora utiliza-se $b = 3$ e $c = -13$, etc. Isto é perfeitamente viável com um computador pequeno, e descobre-se que é possível calcular $kP \pmod{n}$ para todos os valores de b, $b = 3, 4, 5, \ldots, 41$.

Contudo, quando se usa $b = 42$, e $c = -91$, a lei de adição falha e um fator de n é encontrado. O que acontece aqui é o seguinte. Não há problema em fazer uma tabela de $2^i P \pmod{n}$ para $0 \leq i \leq 23$, como foi feito na tabela 1. Então começa-se a somar os pontos na tabela para calcular $kP \pmod{n}$. No penúltimo passo, encontra-se

$$\begin{aligned}(2^4 + 2^6 + 2^{10} + \ldots + 2^{20} + 2^{21})P &= 3863632P \\ &\equiv (1115004543, 1676196055) \pmod{n}.\end{aligned}$$

Calcula-se

$$2^{23}P \equiv (1267572925, 848156341) \pmod{n}.$$

Obtém-se, desta forma, P^k necessário para somar estes dois pontos,

$$(1115004543, 1676196055) + (1267572925, 848156341) \pmod{n}.$$

Para fazer isto deve-se tomar a diferença de suas coordenadas x e encontrar o inverso módulo n. Mas ao tentar fazer isso descobre-se que o inverso não existe porque

$$mdc(1115004543 - 1267572925, n) = mdc(-152568382, 1715761513) = 26927.$$

Então, a tentativa de calcular $12252240(2, 1)$ na curva

$$y^2 = x^3 + 42x - 91 \pmod{1715761513}$$

falha, mas leva à fatoração

$$n = 1715761513 = 26927 \cdot 63719.$$

Pode-se facilmente verificar que cada um destes fatores é primo, logo, esta é a fatoração completa de n.

Neste caso, k foi convenientemente escolhido grande o suficiente. Na prática, se for tomado um b grande, digamos, $b = 1000$, sem que um fator de n seja encontrado, então deve-se aumentar k para $mmc(1, 2, \ldots, 25)$ e começar de novo. Neste caso, provavelmente fará mais sentido usar também um novo ponto inicial, digamos, $P = (3, 1)$.

4 MÉTODOS DE PENEIRAS

Neste capítulo será estudado um método de fatoração utilizando peneiras: o Método de Peneira Quadrática (MPQ), devido a Carl Pomerance ([15]), e será dada uma breve apresentação do Método de Peneira em Extensões Algébricas dos Racionais (NFS)([12]), devido a John Pollard.

MPQ, juntamente com o método das Curvas Elípticas (cap.3), é um dos mais poderosos métodos de fatoração geral, enquanto que NFS é mais usado para fatoração de números com uma forma especial, dos quais o mais famoso é o nono número de Fermat $2^{2^9} + 1 = 2^{512} + 1$, com 155 dígitos.

A idéia básica dos dois métodos é a mesma: por um processo de peneiração procuram-se congruências módulo n trabalhando-se sobre uma base de fatores, e depois faz-se eliminação gaussiana sobre $\mathbb{Z}/n\mathbb{Z}$ para obter uma congruência de quadrados, e assim, uma fatoração de n.

4.1 Método de Peneira Quadrática (MPQ)

Maurice Kraitchik (1882-1957) percebeu que uma grande economia de tempo poderia ser feita se em vez de x e y satisfazendo $x^2 - y^2 = n$ (como no método de Fermat), forem procurados x e y randômicos satisfazendo $x^2 \equiv y^2 \pmod{n}$. Encontrar tal par (x, y) não garante uma fatoração de n. Mas significa que n divide $x^2 - y^2 = (x - y)(x + y)$ e os divisores primos de n estão distribuídos entre os divisores de ambos os fatores: $(x - y)$ e $(x + y)$, de forma que $mdc(n, x - y)$ ou $mdc(n, x + y)$ será um fator não trivial de n, se n não for primo.

Para explicar como a Peneira Quadrática encontra tais inteiros x e y, é mais fácil começar com um algoritmo um pouco mais simples sugerido por John Dixon em 1981.

Escolhe-se aleatoriamente um inteiro r e calcula-se

$$g(r) \equiv r^2 \ (mod\ n).$$

A seguir, fatora-se g(r). Como serão necessários muitos números fatorados, não se deve fazer tentativas demais, por isso escolhe-se um inteiro B que limitará o tamanho dos primos usados na fatoração. Somente serão feitas divisões por tentativas até B, e se não funcionar, escolhe-se um r diferente. Continua-se fazendo isto até que se tenha mais $g(r)'s$ completamente fatorados do que primos abaixo do limite B. Por exemplo, para $B = 10000$, existem 1229 primos menores que B e são necessários, no mínimo, 1230 valores de r para os quais $g(r)$ é fatorado.

Sejam $p_1, p_2, \ldots, p_{1229}$ os primeiros 1229 primos. Se g(r) fatora completamente, então pode ser escrito como

$$g(r) = p_1^{a_1} \cdot p_2^{a_2} \cdot \ldots \cdot p_{1229}^{a_{1229}},$$

onde alguns dos a_i poderão ser zero. A fatoração de $g(r)$ pode ser gravada no vetor

$$v(r) = (a_1, a_2, \ldots, a_{1229}).$$

Se todas as entradas de $v(r)$ são pares, então $g(r)$ é um quadrado perfeito, digamos x^2, e o problema está resolvido, porque

$$x^2 \equiv g(r) \equiv r^2 \ (mod\ n).$$

Infelizmente, este caso é muito improvável. O fato de que o número de $v(r)'s$ é maior do que o comprimento do vetor implica que existe dependência linear entre os vetores. Afirma-se, então, que pode-se encontrar uma soma de $v(r)'s$ distintos na qual todas as entradas são pares. De fato, começa-se estabelecendo

$$w(r) = (b_1, b_2, \ldots, b_{1229}),$$

onde

$$b_i = \begin{cases} 0, & se \quad a_i \quad é \quad par \\ 1, & se \quad a_i \quad é \quad ímpar. \end{cases}$$

Agora, procura-se um subconjunto de r's para os quais a soma dos $v(r)$'s correspondentes tem todas as coordenadas pares, de forma que o produto dos $g(r)$'s correspondentes será um quadrado perfeito. Isto é feito através de eliminação gaussiana módulo 2 na matriz formada pelos vetores $w(r)$. Assim, obtém-se uma congruência da forma

$$g(r_1) \cdot g(r_2) \cdot \ldots \cdot g(r_t) \equiv r_1^2 \cdot r_2^2 \cdot \ldots \cdot r_t^2 \ (mod\ n),$$

onde ambos os lados são quadrados perfeitos. Se isto não fornecer um fator de n, volta-se e encontra-se um subconjunto diferente de $r's$ para os quais os $w(r)$'s são linearmente dependentes módulo 2.

Esta é uma técnica de fatoração probabilística. Não há garantia de que sempre dará um fator de n, pelo menos em algum tempo específico. Mas, na prática, ele extrairá um fator muito grande mais rapidamente do que qualquer algoritmo determinístico.

A eliminação gaussiana, especialmente quando todas as entradas são 0 ou 1, é muito rápida, até mesmo com uma matriz de $1230 \cdot 1230$. O que toma tempo é encontrar aqueles $g(r)'s$ completamente fatorados, porque a maioria dos $r's$ escolhidos randomicamente não fornecerão um $g(r)$ com todos os fatores primos menores que B.

É razoável tratar os valores de $g(r)$ como números aleatórios menores que n, o que significa que a maioria deles serão do mesmo tamanho que n. Espera-se que o maior fator primo de $g(r)$ esteja em torno de $g(r)^{0,63}$([3], pg. 104). Pode-se fatorar $g(r)$ somente se o maior fator primo for menor que B. Se, por exemplo, n (e então $g(r)$) for um inteiro com 25 dígitos, não será possível fatorá-lo, a menos que o maior divisor primo seja aproximadamente $g(r)^{0,16}$. A probabilidade de que isto

aconteça é somente cerca de 1/50000 ([3], pg. 104), o que significa que é preciso escolher cerca de 62 milhões de valores de r a fim de obter 1230 $g(r)'s$ que possam ser fatorados. Portanto, é preciso que se encontre uma maneira de acelerar a escolha dos $g(r)$'s fatoráveis. É aqui que entra a peneira.

4.1.1 Aperfeiçoamento de Pomerance

Para aperfeiçoar este procedimento, Carl Pomerance sugere que seja incorporada uma peneira, como a peneira de Eratóstenes, que permite que se faça fatoração por tentativas em um milhão de números, simultaneamente, sem fazer qualquer divisão.

Ao invés de escolher os $r's$ aleatoriamente, escolhe-se

$$k = \lfloor \sqrt{n} \rfloor.$$

Tomam-se para valores de r: $k+1$, $k+2$,.... Se $f(r)$ for definido como

$$f(r) = r^2 - n,$$

então $f(r) = g(r)$, contanto que $k \leq r \leq \sqrt{2n}$.

Desejam-se encontrar os $f(r)'s$ que podem ser fatorados em primos menores que B. Seja p qualquer primo ímpar menor que B. Assume-se que já foram feitas divisões por tentativas em n até B, então sabe-se que p não divide n. Se p divide $f(r)$, então

$$n \equiv r^2 \ (mod\ p);$$

e o símbolo de Legendre (n/p) é $+1$. Isto significa que serão considerados somente aqueles primos menores que B para os quais $(n/p) = +1$, ou seja, cerca de metade deles. O conjunto de primos pelos quais se tentam dividir os $f(r)'s$ é chamado a **base de fatores**.

Se n é um resíduo quadrático módulo p, digamos, $n \equiv t^2 \ (\mod p)$, e como $n \equiv r^2 \ (\mod p)$, então

$$t^2 \equiv r^2 \ (mod\ p),$$

o que significa que r é congruente a t ou a $-t$ módulo p. Mais importante, se r é congruente a t ou a $-t$ módulo p, então p deve dividir $f(r)$.

Uma vez que for encontrado o primeiro r congruente a t módulo p, sabe-se que $f(r)$ e todo p-ésimo $f(r)$ depois dele será divisível por p (examinando o polinômio $f(r)$, pode-se ver que p também divide $f(r+p),\ f(r+2p),\ etc$). Então encontra-se o primeiro r congruente a $-t$ módulo p, e novamente se percorre a linha. $f(r)$ é o produto de alguns primos da base de fatores, portanto o logaritmo de $f(r)$ é a *soma* dos logaritmos destes primos. Como já se sabe quais $f(r)'s$ são divisíveis por p, sem no entanto fazer qualquer divisão, pode-se apenas armazenar o logaritmo de cada $f(r)$ em uma posição de um vetor, e subtrair[1] o logaritmo de p dos termos apropriados. Isto será feito para cada primo que divide algum $f(r)$. Se o logaritmo restante estiver suficientemente próximo de 0, então o $f(r)$ da posição correspondente foi completamente fatorado.

É claro que um dado primo p pode dividir $f(r)$ mais de uma vez, de forma que se devem resolver também as congruências

$$x^2 \equiv n\ (mod\ p^a)$$

onde p é um primo ímpar e o expoente em p aumenta rapidamente para cerca de

$$\frac{2logL}{logp},$$

([3], pg. 105) onde L é o maior primo na base de fatores.

Na prática, entretanto, isto significa que uma maior seleção (peneiração) está sendo feita nos primos pequenos. Há dois problemas em peneirar primos pequenos. Primeiro, é lento. Quando $p = 3$ subtrai-se $log\ 3$ de toda terceira entrada. Quando $p = 311$, uma peneira simples tomará menos que 1 centésimo do tempo porque subtrai-se $log\ 311$ de toda $311^{\underline{a}}$ entrada. Segundo, $log\ 3 = 1.098...$ é muito menor que $log\ 311 = 5.739....$

[1] Ao invés de armazenar o logaritmo de $f(r)$ e subtrair o logaritmo de p dos termos apropriados, pode-se preencher o vetor com zeros e somar o logaritmo de p aos termos apropriados. Se o logaritmo restante estiver suficientemente próximo do logaritmo de $f(r)$, então o $f(r)$ da posição correspondente foi completamente fatorado.

Resumindo, há três passos no uso da Peneira Quadrática:

- Encontrar uma base de fatores e resolver a congruência

$$x^2 \equiv n \ (mod \ p)$$

para cada primo p na base de fatores.

- Executar a operação de peneiração para encontrar $f(r)'s$ suficientes os quais podem ser completamente fatorados sobre a base de fatores.

- Usar eliminação gaussiana para encontrar um produto dos $f(r)'s$ que é um quadrado perfeito.

4.1.2 Resolução de Congruências Quadráticas

Para cada primo ímpar p na base de fatores, é preciso resolver a congruência

$$x^2 \equiv n \ (mod \ p) \tag{4.1}$$

onde n é um resíduo quadrático módulo p.

A base de fatores utilizada deve ser grande o suficiente para que haja uma probabilidade razoável de que um dado $f(r)$ será fatorado. No entanto, há a necessidade de manter a base de fatores pequena o suficiente para que se possa fazer eliminação gaussiana em uma matriz cujas dimensões são o tamanho da base de fatores.

Deseja-se encontrar, para cada primo da base de fatores, uma potência maior que divida $r^2 - n$. Para começar, o primo 2, que é um primo excepcional, será analisado. É claro que o número n a ser fatorado é ímpar. Assim, para $n \equiv q \ (mod \ 8)$ existem quatro valores possíveis para q :1, 3, 5 e 7. Se n é congruente a 3 ou 7 módulo 8, então $r^2 - n$ é divisível por 2 quando r é ímpar e nunca é divisível por qualquer potência mais alta de 2. Pode-se ver isto nos cálculos abaixo.

Seja $n \equiv 3 \ (mod \ 8)$, ou seja, $n = 8k + 3$ para $k = 0, 1, \ldots$.

(1) Seja r par, isto é, $r = 2m$. Então

$$\begin{aligned} r^2 - n &= (2m)^2 - (8k+3) \\ &= 4m^2 - 8k - 3 \\ &= 2(2m^2 - 4k - 1) - 1 \\ &= 2q - 1. \end{aligned}$$

Desta forma, conclui-se que $r^2 - n$ não é divisível por nenhuma potência de 2 se r é par.

(2) Seja r ímpar, isto é, $r = 2m + 1$. Então

$$\begin{aligned} r^2 - n &= (2m+1)^2 - (8k+3) \\ &= 4m^2 + 4m + 1 - 8k - 3 \\ &= 2(2m^2 + 2m - 4k - 1) \\ &= 2c, \end{aligned}$$

o que comprova que $r^2 - n$ é divisível por 2 quando r é ímpar (e nunca por uma potência mais alta de 2).

Da mesma forma, se n é congruente a 5 módulo 8, então $r^2 - n$ é divisível por 4 quando r é ímpar, mas nunca é divisível por 8. Se n é congruente a 1 módulo 8, então $r^2 - n$ é divisível por 8, no mínimo, sempre que r é ímpar.

Deseja-se, obviamente, ter n congruente a 1 módulo 8. Quando isto acontece, pode-se diminuir $log\ 8$, em vez de $log\ 2$, de cada $f(r)$ que é divisível por 2. Entretanto, se n não é congruente a 1 módulo 8, introduz-se um **multiplicador**, substituindo n por um de seus múltiplos a fim de aumentar as chances de que n fatorará completamente. Se um primo divide seu multiplicador, então ele dividirá $r^2 - n$ se e somente se dividir r. Deve-se colocar o multiplicador antes de encontrar a base de fatores. Por exemplo, se o número dado para fatorar for congruente a 5 módulo 8, então será multiplicado por 5, se for congruente a 3 módulo 8 então será multiplicado por 3, e se for congruente a 7 módulo 8 então será multiplicado por 7. Deve-se lembrar que a Peneira Quadrática não encontra fatores de um dado

tamanho mais rápido que os de outro, de forma que rodar a Peneira Quadrática em $3n$, por exemplo, não diminui suas chances de encontrar um fator grande de n.

A fim de ilustrar as explicações acima, segue-se um exemplo numérico.

4.1.3 Exemplo

Seja $n = 8051$ o inteiro a ser fatorado. Em geral, o MPQ não é utilizado para fatorar números com apenas quatro dígitos, mas este pequeno valor de n, o qual já foi usado como exemplo do método Rho, serve para mostrar como funciona o MPQ.

A base de fatores escolhida tem 10 números primos, e foram usados 200 valores de r. Começa-se calculando $k = \lfloor \sqrt{n} \rfloor$. Em vez de fazer $r = k + i$, para $i = 1, 2, \ldots, b$, pode-se fazer

$$k - \frac{b}{2} < r < k + \frac{b}{2},$$

ou seja, tomam-se os valores de r em torno da raiz quadrada de n. Se acontecer de $f(r)$ ser um número negativo, pode-se usar -1 juntamente com os primos da base (embora -1 não seja primo). Neste exemplo, onde $k = \lfloor \sqrt{8051} \rfloor = 89$, usa-se

$$r = 89 + i, \ para \ i = 1, 2, \ldots, 200,$$

de forma que (-1) não pertence à base.

Para cada primo p na base de fatores, n deve ser um resíduo quadrático módulo p, ou seja, deve satisfazer

$$(n/p) = +1.$$

A base obtida para o exemplo é:

$$5 \quad 7 \quad 13 \quad 23 \quad 43 \quad 47 \quad 59 \quad 61 \quad 79 \quad 103.$$

Para cada um destes primos, é preciso resolver a congruência

$$n \equiv t^2 \ (mod \ p).$$

A menor solução positiva t é dada na posição do primo correspondente. Não se pode esquecer que $p - t$ é outra solução.

$$1 \quad 1 \quad 2 \quad 1 \quad 15 \quad 22 \quad 26 \quad 11 \quad 25 \quad 29.$$

Ao invés de calcular o logaritmo do valor absoluto de $f(r)$ duzentas vezes, Silverman sugere que se comece com um vetor de zeros ao qual soma-se $log\ p$ quando p dividir o $f(r)$ correspondente. Para cada primo p da base, o primeiro $f(r)$ ao qual será somado $log\ p$ é indicado pelo t correspondente. A seguir, soma-se $log\ p$ a cada p-ésimo $f(r)$ depois do primeiro. A menos que p divida o multiplicador de n, há duas soluções da equação (4.1). Então também se deve somar $log\ p$ aos $f(r)$'s começando em $p - t$ e a cada p-ésimo $f(r)$ depois dele. Se forem usados $2M$ valores para r, então o valor absoluto de $(\lfloor\sqrt{n}\rfloor - M + i)^2 - n$ será aproximadamente

$$TARGET = (log\ n)/2 + log\ M.$$

Quando a peneira é feita, há suficientemente poucas entradas próximas a $TARGET$ para as quais se pode fazer divisão por tentativas sobre a base de fatores para ver exatamente quais fatoram completamente.

Mas deve-se saber quanto é suficientemente próximo. Se a parte não fatorada restante é menor que o quadrado do maior fator primo da base, então ela é prima. Ainda que não se tenha uma fatoração completa sobre a base de fatores, obtém-se uma fatoração que ainda poderá ser usada. A sugestão de Silverman, portanto, é estabelecer

$$CLOSENUF = TARGET - T \cdot log\ (pmax)$$

onde $pmax$ é o maior primo na base de fatores e T é uma constante próxima de 2.

Esta modificação feita por Silverman significa que serão perdidos alguns valores r para os quais $f(r)$ fatora completamente, mas a aceleração da peneira compensa.

No exemplo, foram encontrados 11 $f(r)$'s completamente fatorados, que são os seguintes:

$$
\begin{array}{rrl}
i = & 413 & f(r) = 7 \cdot 59 \\
& 2765 & 5 \cdot 7 \cdot 79 \\
& 3185 & 5 \cdot 7^2 \cdot 13 \\
& 4945 & 5 \cdot 23 \cdot 43 \\
& 5405 & 5 \cdot 23 \cdot 47 \\
& 9373 & 7 \cdot 13 \cdot 103 \\
& 12685 & 5 \cdot 43 \cdot 59 \\
& 34385 & 5 \cdot 13 \cdot 23^2 \\
& 36049 & 13 \cdot 47 \cdot 59 \\
& 60593 & 13 \cdot 59 \cdot 79 \\
& 73745 & 5 \cdot 7^3 \cdot 43
\end{array}
$$

Para encontrar qual a combinação apropriada de $f(r)$'s que é um quadrado perfeito, usa-se eliminação gaussiana. Para cada $f(r)$ fatorado associa-se um *string* de 10 dígitos binários, cada coluna correspondendo a um dos primos da base de fatores. Se o primo correspondente tem potência par, o dígito que o representa é 0; se a potência é ímpar, o dígito é 1.

Foram encontrados 11 $f(r)$'s completamente fatorados, cada um representado por um *string* de 10 dígitos. Isto dá origem a uma matriz B de 0's e 1's, com 11 linhas e 10 colunas.

$$\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Uma matriz identidade $I_{11\times 11}$ é associada à matriz B a fim de dizer qual a combinação de $f(r)$'s dará um quadrado perfeito. Todos os passos da eliminação gaussiana feitos na matriz B devem ser feitos na matriz identidade. A eliminação gaussiana é feita até que se encontre uma linha de zeros em B, e a linha correspondente da matriz I dirá quais $f(r)$'s devem ser multiplicados para que se encontre um quadrado perfeito. Por exemplo, a oitava linha de B foi a primeira a ser zerada. A linha correspondente de I, seguindo os mesmos passos, é

$$[0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0].$$

As colunas onde aparece o dígito 1 indicam quais $f(r)$'s devem ser multiplicados. Neste caso, o terceiro e o oitavo.

Multiplicando as correspondentes decomposições primas, obtém-se

$$(11236)^2 \cdot (42436)^2 \equiv 5^2 \cdot 7^2 \cdot 13^2 \cdot 23^2 \ (mod\ 8051).$$

Esta congruência é da forma

$$x^2 \equiv y^2 \ (mod\ n).$$

Agora, deve-se calcular x e y módulo n, e a seguir, o $mdc(x-y,n)$. Se o mdc for 1 ou n, deve-se procurar a próxima linha zerada, ou caso todas elas já tenham sido usadas, escolher uma nova base de fatores e novos $f(r)$'s. Se $1 < mdc(x-y,n) < n$, então um fator de n foi encontrado.

No exemplo, $x = 10465$, $y = 21836$ e $mdc(x-y,n) = mdc(11371, 8051) = 83$, ou seja, $8051 = 83 \cdot 97$.

Até mesmo com um milhão de entradas a peneira funciona muito rapidamente. O tempo usado para encontrar as soluções e peneirá-las é muito menor do que para rodar divisão por tentativas um milhão de vezes.

4.1.4 Aperfeiçoamento na Resolução de Congruências Quadráticas

Resolver a congruência quadrática (4.1) para cada primo na base de fatores toma bastante tempo. No entanto, existem alguns critérios que aceleram essa etapa, pois de acordo com a forma de cada primo usado, encontram diretamente o valor de x sem que seja preciso resolver a congruência. Tais critérios são dados na forma de dois teoremas: Teorema 4.1.1, que é válido para os casos onde $p \equiv 3 \ (mod\ 4)$ ou $p \equiv 5 \ (mod\ 8)$, e Teorema 4.1.2, válido para qualquer primo ímpar, mas um pouco mais lento que os procedimentos descritos no Teorema 4.1.1.

Teorema 4.1.1. *Se n é um resíduo quadrático módulo o primo p e*

- *se $p = 4k + 3$, então*

$$x \equiv n^{k+1} \ (mod\ p),$$

  é uma solução da equação (4.1).

- *Se $p = 8k + 5$ e $n^{2k+1} \equiv 1 \ (mod\ p)$, então*

$$x \equiv n^{k+1} \ (mod\ p),$$

  é uma solução da equação (4.1).

- *Se* $p = 8k + 5$ *e* $n^{2k+1} \equiv -1 \ (mod\ p)$, *então*

$$x \equiv (4n)^{k+1} \cdot (\frac{p+1}{2}) \ (mod\ p),$$

é uma solução da equação (4.1).

Prova: Visto que n é um resíduo quadrático mod p, sabe-se que $n^{(p-1)/2} \equiv 1 \pmod{p}$.
Se $p = 4k + 3$ então

$$(n^{k+1})^2 = n^{2k+2} = n \cdot n^{(p-1)/2} \equiv n \ (mod\ p).$$

Se $p = 8k + 5$, então

$$n^{4k+2} \equiv n^{(p-1)/2} \equiv 1 \quad (mod\ p),$$

o que implica que

$$n^{2k+1} \equiv 1\ ou\ \ -1 \ (mod\ p).$$

Se $n^{2k+1} \equiv +1 \ (\mod p)$ então

$$(n^{k+1})^2 = n^{2k+1} \cdot n \equiv n \ (mod\ p).$$

Se $n^{2k+1} \equiv -1 (\mod p)$, então

$$(4n)^{2k+2} = 2^{4k+2} \cdot n^{2k+2} = 2^{4k+2} \cdot n^{2k+1} \cdot n$$

$$\equiv -1 \cdot (-n)(mod\ p),$$

porque 2 não é um resíduo quadrático módulo p.

Teorema 4.1.2. *Seja* n *um resíduo quadrático módulo um primo* p *e seja* h *escolhido de forma que o símbolo de Legendre* $(h^2 - 4n/p)$ *é -1. Define-se uma seqüência* $v_1, v_2, \ldots$ *pela recursão*

$$\begin{aligned} v_1 &= h, \\ v_2 &= h^2 - 2n, \\ v_3 &= h \cdot v_{i-1} - h \cdot v_{i-2}. \end{aligned}$$

Temos, então, que

$$\begin{aligned} v_{2i} &= v_i^2 - 2n^i, \\ v_{2i+1} &= v_i \cdot v_{i+1} - h \cdot n^i, \end{aligned}$$

e uma solução da equação (4.1) é dada por

$$x \equiv v_{(p+1)/2} \cdot \left(\frac{p+1}{2}\right) \ (mod\ p).$$

Este algoritmo foi sugerido por D.H.Lehmer em 1969, e sua prova pode ser vista em ([3], pg. 187-188).

Nota-se que um h satisfatório pode ser encontrado aleatoriamente testando-se diferentes valores, então isto não tomará muito tempo.

A relação entre v_{2i} e v_i pode ser usada para calcular um v_j arbitrário em aproximadamente $\log j$ passos da mesma forma que o algoritmo que calcula $a^b \ (mod\ m)$ exponencia em tempo proporcional ao logaritmo do expoente. Conhecendo v_i e v_{i+1}, pode-se calcular v_{2i},v_{2i+1} e v_{2i+2}. Qual dos v's devem ser mantidos: dependendo da expansão binária de j, v_{2i} e v_{2i+1} ou v_{2i+1} e v_{2i+2}.

Refinamentos da Peneira Quadrática

Dois refinamentos da Peneira Quadrática, entre os muitos que foram sugeridos e implementados, mostraram-se particularmente úteis na redução do tempo computacional: o uso de primos grandes e o uso de polinômios múltiplos.

4.1.5 Refinamento Usando Primos Grandes

Se a peneira é executada, por exemplo, para o inteiro $n = 4999486012441$ com 13 dígitos e 10000 valores de $f(r)$, são encontrados 138 valores de i para os quais chega-se até divisão por tentativas. Somente 39 destes fatoram completamente sobre uma base de 30 fatores escolhida. Os outros 99 fatoram como um produto de primos da base vezes um fator adicional menor que $397^{1.5}$ (e então o fator extra é

necessariamente um primo). O refinamento de primos grandes usa estas fatorações extras que envolvem primos grandes.

Há uma boa chance de que vários primos grandes aparecerão mais que uma vez nestas fatorações extras. Para $n = 4999486012441$, se for usada peneira sobre o intervalo de comprimento 8000, encontram-se 32 valores de $f(r)$ que fatoram completamente sobre a base. Entretanto, também se obtém três fatorações que envolvem o primo 449, que não pertence à base. Isto significa que o produto de quaisquer dois destes três $f(r)$'s será um produto de primos na base de fatores vezes um quadrado perfeito (ou seja, 449^2). Como o objetivo é encontrar um produto dos $f(r)'s$ que é um quadrado perfeito, este produto envolvendo o primo grande é tão bom quanto uma fatoração completa sobre a base de fatores.

Somente dois dos três produtos possíveis são mantidos das correspondentes seqüências de trinta dígitos que somarão o terceiro. Há também duas fatorações que envolvem 443 e duas fatorações que incluem o fator 1097. Então, o uso de primos grandes acrescenta quatro fatorações aos 32 $f(r)$'s que fatoram completamente sobre a base de fatores. Com um total de 36 vetores de 30 dígitos, obtém-se pelo menos seis produtos distintos de $f(r)'s$ que são quadrados perfeitos.

4.1.6 Refinamento Usando Polinômios Múltiplos

O segundo refinamento foi sugerido por Peter Montgomery ([13]). Os $f(r)'s$ são menores, e então mais próprios para fatorar, quando r está próximo à raiz quadrada de n. No exemplo da peneira de comprimento 10000, dezesseis dos trinta e nove $f(r)$'s que fatoram completamente sobre a base de fatores têm um r que está a uma distância de até 1000 da raiz quadrada de 4999486012441. O refinamento de Montgomery peneira sobre um intervalo mais curto mas com vários polinômios quadráticos diferentes em r. Ao invés de apenas

$$f(r) = r^2 - n,$$

consideram-se polinômios da forma

$$F(r) = ar^2 + 2br + c.$$

Se $n = b^2 - ac$, então

$$\begin{aligned} a \cdot F(r) &= a^2r^2 + 2abr + ac \\ &= a^2r^2 + 2ab + b^2 - n \\ &= (ar + b)^2 - n. \end{aligned}$$

Como antes, se um primo p divide $a \cdot F(r)$, então n é um resíduo quadrático módulo p, de forma que não é preciso mudar a base de fatores usada anteriormente.

O número a ser fatorado atinge seu mínimo em $r = -b/a$. Deseja-se escolher a, b e c de forma a minimizar ambos:

$$-F(-b/a) = n/a$$

e os valores extremos no extremo do intervalo peneirado

$$F(-M - b/a) = F(M - b/a) = a \cdot M^2 - n/a.$$

Se M é predeterminado, isto é realizado estabelecendo-se estes valores iguais, isto é, a deveria ser cerca de

$$\frac{\sqrt{2n}}{M}.$$

Se o a escolhido é primo, então sabe-se como resolver a congruência

$$x^2 \equiv n \ (mod \ a).$$

Escolhe-se b como uma solução desta congruência e c como sendo

$$c = (b^2 - n)/a.$$

A peneira quadrática multipolinomial descrita acima tem muitas características interessantes. O limite superior no valor de $F(r)$ é menor que o limite em

$f(r)$, de forma que a chance de fatorar completamente os números é maior. O intervalo usado para fazer a peneiração pode ser encurtado. Se não for obtido um número suficiente de $F(r)$'s completamente fatorados, então gera-se um novo polinômio e peneira-se novamente sobre o intervalo mais curto, pois manter o intervalo curto aumenta as chances de que um dado $F(r)$ fatorará. A peneira paraleliza perfeitamente. Usando N processadores, pode-se determinar um polinômio diferente para cada processador e o algoritmo roda N vezes mais rápido.

4.2 Método de Peneiras em Extensões Algébricas dos Racionais (NFS)

O método NFS, apresentado em 1988 por John Pollard, até o momento parece ser o algoritmo para fatoração de inteiros assintoticamente mais rápido conhecido. Há duas formas de NFS: o NFS Especial, usado para fatorar inteiros na forma $n = r^e - s$, onde r é inteiro positivo pequeno e s é um inteiro não nulo de valor absoluto pequeno; e o NFS Geral, aplicado a números arbitrários. NFS tem um tempo de execução heurístico ([11],[12])

$$e^{(O((log\ n)^{1/3}(log\ log\ n)^{2/3}))}$$

assintoticamente muito melhor que o tempo ([11],[12])

$$e^{(O(log\ n\ log\ log\ n)^{1/2})}$$

tomado pelo MPQ, o melhor algoritmo para fatoração para inteiros com menos de 105 dígitos, aproximadamente. Mas, na prática, o NFS é, atualmente, o algoritmo mais rápido para fatoração de inteiros com mais de 150 dígitos.

Seja n um número ímpar a ser fatorado. Da mesma forma que o MPQ, o NFS tenta encontrar uma solução da equação $x^2 \equiv y^2\ (mod\ n)$. Para pelo menos metade dos pares $(x\ mod\ n, y\ mod\ n)$ com $x^2 \equiv y^2\ (mod\ n)$ e x e y relativamente primos a n, o $mdc(x-y,n)$ dá um fator não trivial de n. O NFS usa uma base de fatores no anel de inteiros de uma extensão algébrica adequadamente escolhida.

Segue-se uma breve descrição de alguns passos básicos do NFS.

Para começar, deve-se escolher um polinômio $f(x) \in \mathbb{Z}[x]$ mônico e irredutível, e faz-se $n = f(m)$, para algum $m \in \mathbb{Z}$. O grau do polinômio é importante; para números maiores usa-se $d = 5$ ou 6, e para números menores, $d = 3$ ou 4. A seguir, deve-se determinar a base de fatores, cujo comprimento depende do tamanho de n. O próximo passo é fazer a peneira, ou seja, testar para tantos pares (a, b) quanto possível, tais que $a + b$ e

$$b^d f(-a/b) = (-a)^d + a_{d-1}(-a)^{d-1} + a_{d-2}(-a)^{d-2}b^2 + \ldots - a_1 a b^{d-1} + a_0 b^d$$

são divisíveis somente pelos primos da base de fatores, ou seja, são suaves sobre a base. Para que se obtenha sucesso, a proporção de números suaves entre os valores do polinômio deveria ser aproximadamente a mesma que a proporção de números suaves entre todos os números do mesmo tamanho. Existem mais alguns passos intermediários até que se chegue a $x^2 \equiv y^2 \ (mod\ n)$. Então calcula-se $mdc(x - y, n)$ para obter o divisor de n.

5 CONCLUSÃO

A fatoração de números inteiros, um assunto muito antigo que tem fascinado os matemáticos por séculos é atualmente, de interesse não só acadêmico, mas de todos aqueles envolvidos em transmissão segura de dados via computador. Dentre os motivos que geram este aumento de interesse pode-se citar a crescente utilização dos criptossistemas de chave pública, cuja segurança se baseia na ineficiência dos métodos de fatoração.

Os algoritmos modernos para fatorar inteiros caem em duas categorias. A primeira engloba os métodos de divisão por tentativas, Rho, $p-1$ e o das curvas elípticas, os quais encontram fatores primos pequenos rapidamente. Os algoritmos da segunda categoria, dentre os quais estão o de Peneira Quadrática(MPQ) e o de Peneiras em Extensões Algébricas dos Racionais(NFS), fatoram um número sem levar em conta o tamanho de seus fatores primos, mas são muito mais caros quando aplicados a inteiros grandes. O principal objetivo deste trabalho foi o estudo de tais métodos de fatoração.

No capítulo 1, estudou-se o método de divisão por tentativas, o qual procura fatores pequenos em ordem crescente até, no máximo, a raiz quadrada de n. Por ser o algoritmo mais elementar, é o primeiro método a ser usado quando se deseja fatorar um inteiro, embora seja ineficiente (no pior caso, quando n é primo, ele terá que executar o maior número de laços, $\sqrt{n}$ laços). Para inteiros muito grandes, deve ser usado para fatores de até cerca de 6 dígitos. Se não funcionar, tenta-se um dos métodos de Pollard, por exemplo.

No capítulo 2 foram apresentados três métodos elementares: o de Fermat, e os métodos Rho e $p-1$ de Pollard.

O método de Fermat funciona na direção oposta à da divisão por tentativas, ou seja, começa procurando fatores próximos à raiz quadrada de n e continua de forma decrescente. Atualmente, este algoritmo é pouco implementado a menos

que se saiba que o número a ser fatorado é divisível por dois primos relativamente próximos à sua raiz quadrada. A importância deste método se deve principalmente ao fato de que ele serve de base para um dos mais poderosos algoritmos de fatoração, o MPQ.

Os métodos de Pollard são chamados algoritmos probabilísticos, e são usados para encontrar fatores pequenos de inteiros grandes. O método $p-1$ pode encontrar um fator p muito rapidamente se $p-1$ for composto de pequenos primos. Essa é uma das razões para as restrições nos primos p e q no criptossistema RSA, pois se $p-1$ ou $q-1$ tem somente fatores primos pequenos, então o método $p-1$ quebrará o código muito rapidamente. Em 2001 Cage e Woltman encontraram um fator de 42 dígitos $p = 2262111246861207828352339453442536710495 43$ (o maior encontrado pelo método $p-1$) do inteiro $n = 26659-1$, com $p-1 = 2.3^5.7.89.827.6659.10177.109197973.178131337.685331099$.

Uma vez que já se tenha usado divisão por tentativas para obter todos os pequenos divisores do número, mas ele ainda é composto, e percebe-se que já esgotaram as possibilidades dos algoritmos Rho e $p-1$ de Pollard, então é preciso considerar um método mais poderoso: MCE ou MPQ.

O MCE (cap. 3) atualmente é o melhor algoritmo conhecido para encontrar fatores não tão grandes de números compostos, ou seja, fatores com cerca de 50 dígitos. Seu tempo de execução depende do tamanho do menor divisor primo p de N, e não do próprio N. Em 26 de de dezembro de 1999 Nyk Lygeros e Michel Mizony, dois pesquisadores matemáticos de Lyon (França), encontraram um fator primo de 54 dígitos de um número composto de 127 dígitos usando GMP-ECM, uma implementação livre do MCE. De acordo com a tabela mantida por Richard Brent [23] este é o maior fator primo já encontrado usando MCE. O recorde anterior era de um primo de 53 dígitos encontrado em setembro de 1998 por Conrad Curry.

O MPQ , usado para fatorar inteiros grandes que não tenham fatores primos significantemente menores que $\sqrt{n}$, é o melhor algoritmo para fatoração de

inteiros com menos de 105 dígitos, aproximadamente. O MPQ foi usado para fatorar o RSA-129, um inteiro de 129 dígitos. Em agosto de 1977 Martin Gardner descreveu o código do RSA em sua coluna *Scientific American*. Para demonstrar o poder do RSA, os pesquisadores do M.I.T. codificaram uma mensagem, usando um inteiro n com 129 dígitos, que ficou conhecido como RSA-129, e um número e com 4 dígitos. O texto codificado resultante foi publicado por Gardner, e os pesquisadores do M.I.T. ofereceram \$100 para a primeira pessoa que quebrasse o código, pois esperavam que isto levasse em torno de 23000 anos. No entanto, o código foi quebrado em apenas 17 anos, em abril de 1994. Um grupo de 600 voluntários, espalhados por vários países, encontraram os dois fatores de 64 e 65 dígitos do RSA-129 em apenas oito meses, utilizando para isto o MPQ.

O NFS é o algoritmo mais rápido para fatoração de inteiros com mais de 150 dígitos. Este método foi usado para fatorar o 9º número de Fermat, $F_9 = 2^{512}+1$, com 155 dígitos decimais. O inteiro F_9 tem três fatores com 7, 49 e 99 dígitos, sendo que os dois últimos foram encontrados em 1999 através do NFS. Este método também foi usado na fatoração do RSA-130 (abril de 1996), do RSA-140 (fevereiro de 1999) e RSA-155 (agosto de 1999).

Fatoração de números inteiros é um assunto que nunca deixará de ser atual. Novas aplicações são descobertas a cada dia que passa, fazendo com que não só os matemáticos, mas também pessoas de outras áreas, tenham interesse no estudo constante deste assunto, na procura de um único método capaz de fatorar inteiros de qualquer tamanho de forma rápida e eficiente.

BIBLIOGRAFIA

[1] Brent, R.P., *Some Integer Factorisation Algorithms using Elliptic Curves*, Australian Computer Science Communications 8, 1986, 149-163. Retyped, with corrections and postscript, by Frances Page at Oxford University Computing Laboratory, 1998.

[2] Brent, R.P., *Some Parallel Algorithms for Integer Factorisation*, Expanded Version of Invited Paper to be presented at Proc. Europar'99, Toulouse, 1999. Short Version in LNCS **1685**, Springer-Verlag, Berlin, pp. 1-22.

[3] Bressoud, D.M., *Factorization and Primality Testing*, Springer-Verlag, 1998.

[4] Cohen, H., *A Course in Computational Algebraic Number Theory*, Springer-Verlag, Berlin Heidelberg, 1993.

[5] Coutinho, S.C., *Números Inteiros e Criptografia RSA*, Rio de Janeiro, IMPA/SBM, 1997.

[6] Guy, R.K., *How to Factor a Number*,Proceedings of the Fifth Manitoba Conference on Numerical Mathematics, 1975, pp. 49-89.

[7] Koblitz, N., *A Course in Number Theory and Criptography*, 2ª ed., Springer-Verlag, New York, 1994.

[8] Knuth,D.E., *Seminumerical Algorithms*, Vol. 2 de *"The Art of Computer Programming"*, 2ª ed., Addison-Wesley Publishing Company, Reading, Massachussets, 1981.

[9] Landau, E., *Elementary Number Theory*, Chelsea, New York, 1958.

[10] Lenstra Jr., H.W., *Factoring Integers using Elliptic Curves*, Annals of Mathematics, (1987), pp. 649-673.

[11] Lenstra, A.K, Lenstra Jr., H.W., *The Development of the Number Field Sieve*, Lecture Notes in Mathematics, volume 1554, 1993, Springer-Verlag.

[12] Lenstra, A.K, Lenstra Jr., H.W., Manasse, M.S., Pollard, J.M., *The Number Field Sieve*, ACM Symposium on Theory of Computing, 1990, pp. 564-572.

[13] Montgomery, P.L., *A Survey of Modern Integer Factorization Algorithms*, CWI Quarterly **7**, 1994, pp. 337-366.

[14] Ore, O., *An Invitation to Number Theory*, Random House, New York, 1967.

[15] Pomerance, C., *The Quadratic Sieve Factoring Algorithm*, Lecture Notes in Computer Science, Vol. 209, pp. 169-182, Springer-Verlag, 1985.

[16] Pomerance, C., *Factoring*, Proceedings of Symposia in Applied Mathematics, Vol.42, pp. 27-47, American Mathematical Society, 1990.

[17] Rosen, K.H., *Elementary Number Theory and its Applications*, 3ª ed., Addison-Wesley Publishing Company, New Wesley, 1993.

[18] Santos, J.P.O, *Introdução à Teoria dos Números*, Rio de Janeiro, IMPA, CNPq, 1998.

[19] Sethuraman, B.A., *Rings, Fields, and Vector Spaces*, Springer-Verlag, New York, 1997.

[20] Silverman, J.H.,Tate, J., *Rational Points on Elliptic Curves*, Springer-Verlag, New York, 1992.

[21] Silverman, J.H., *A Friendly Introduction to Number Theory*, Prentice-Hall, 1997.

[22] Silverman, J.H., *The Arithmetic of Elliptic Curves*, Springer-Verlag, 1986.

[23] ftp://ftp.comlab.ox.ac.uk/pub/Documents/techpapers/Richard.Brent/champs.ecm