IMPLEMENTAÇÃO DO MÉTODO DE FATORAÇÃO DE INTEIROS CRIVO QUADRÁTICO

Aline de Paula Sanches[1]; Adriana Betânia de Paula Molgora[2]

[1]Estudante do Curso de Ciência da Computação da UEMS, Unidade Universitária de Dourados; E-mail: alinee_sanches@hotmail.com

[2]Professora do Curso de Ciência da Computação da UEMS, Unidade Universitária de Dourados; E-mail: abmol@terra.com.br

Área Temática: Teoria Computacional dos Números

Resumo. O Crivo Quadrático é considerado como um dos métodos de fatoração de inteiros mais importantes da atualidade. No entanto, implementar este método não é uma tarefa trivial, pois envolve estudos sobre aspectos matemáticos e computacionais envolvidos em seu funcionamento. O objetivo deste trabalho é apresentar uma descrição de uma implementação realizada durante o desenvolvimento do mesmo.

Palavras-chave: Algoritmo.Congruência. Diferença de quadrados.

1. Introdução

O problema de fatoração de números inteiros tem ocupado lugar de destaque na Teoria dos números. Essa importância deve-se ao fato de que sistemas criptográficos como o RSA têm sua segurança baseada na dificuldade de fatorar um número inteiro qualquer dado, afirma Coutinho (2000).

O método de fatoração de inteiros Crivo Quadrático, devido a Pomerance (1985, p.169), é considerado como um dos métodos de fatoração mais potentes, sendo utilizado na fatoração de números de aproximadamente 100 dígitos decimais.

Esse trabalho tem como objetivo realizar um estudo prático do método Crivo Quadrático, apresentando uma descrição de uma implementação realizada durante o desenvolvimento do mesmo, e a análise dos resultados obtidos.

2. Materiais e Métodos

Para o alcance do objetivo proposto, foram realizados as seguintes etapas:

1. Estudo do manual da biblioteca GMP;
2. Integração da GMP em um compilador C;

3. Estudo e implementação do método de fatoração de inteiros Crivo Quadrático, utilizando as funções da biblioteca GMP;

4. Documentação do processo de implementação do método Crivo Quadrático, para um maior entendimento.

3. Resultados e Discussão

3.1. Crivo Quadrático

A fatoração de inteiros através do método Crivo Quadrático tem como base o fato de que se existirem números x e y que satisfaçam a condição $x^2 \equiv y^2 (\mathrm{mod}\, n)$, então tem-se que $(x+y)\cdot(x-y) \equiv 0 \ (\mathrm{mod}\, n)$. Logo, $\frac{n}{(x^2-y^2)} = (x+y)\cdot(x-y)$ e os números $d = mdc(x+y, n)$ e $f = mdc(x-y, n)$ poderão ser fatores não triviais de *n*. Ou seja, a idéia básica do método consiste em encontrar congruências da forma $r_i^{\,2} \equiv y_i (\mathrm{mod}\, n)$, onde $\prod y_i = y^2$ é um quadrado perfeito. Se $x = \prod r_i$, então $x^2 \equiv y^2 (\mathrm{mod}\, n)$.

De acordo com Crandall e Pomerance (2002, p.01), na prática, para encontrar *x* e *y*, em primeiro lugar deve-se encontrar uma base de fatores, que é um conjunto de números primos como, por exemplo, o conjunto $\{-1, 2, p_2, ..., p_k\}$, tal que $p_i \le B$, para um certo limite B e, para cada primo *p*, o número *n* deve ser um resíduo quadrático módulo p. Em seguida, são calculados números $f(r_i)$'s dados por $f(r_i) = r_i^{\,2} - n$ para r_i próximo de $\sqrt{n}$.

Num segundo momento, devem-se determinar r_i's suficientes para os quais $f(r_i)$ pode ser completamente fatorado pela base de fatores. A quantidade desses $f(r_i)$'s deve ser maior do que o número de primos e menores do que *B*.

Armazenando os $f(r_i)$'s , em um vetor na base binária, utiliza-se a adição de vetores para descobrir uma combinação linear que produz um vetor nulo que corresponderá a um quadrado perfeito. Então *x* será dado pelo produto dos r_i's correspondentes módulo *n*, e *y* será dado pela raiz do produto dos fatores dos $f(r_i)$'s correspondentes. Em seguida é calculado $d = mdc(x+y, n)$. Se *d* é fator não trivial de *n*, então um fator foi encontrado e, para determinar o segundo fator basta calcular a divisão de *n* por *d*.

Em resumo, pode-se dizer que os passos para a fatoração de *n* pelo método Crivo Quadrático, são:

1º Encontrar uma base de fatores.

2º Determinar um conjunto de números que podem ser completamente fatorados sobre a base de fatores.

3º Usar a Eliminação Gaussiana para encontrar um produto dos números determinados no 2º passo que seja um quadrado perfeito.

Algoritmo (Crivo Quadrático)

Entrada: *n*, B *// número a ser fatorado e limite B//*
Saída: *fator1, fator2* *//fatores de n//*
[Início]
//Nessa etapa é determinada a base de fatores e calculado o símbolo de Legendre//
$p_1 = 2$, $r_1 = 1$;
Para $2 \le i \le k$ encontre números primos $p_i \le B$ tais que $\left(\frac{n}{p}\right) = 1$;
Para $2 \le i \le k$ encontre as raízes $\pm r_i$ com $r_i^2 \equiv n \pmod{p_i}$;

Determine um conjunto S de $k+1$ pares $(r_i, r_i^2 - n)$ onde r é dado pela seqüência de números $r = \lfloor \sqrt{n} \rfloor, \lfloor \sqrt{n} \rfloor + 1, \ldots$ e $(r^2 - n)$ pode ser totalmente fatorado pela base de fatores;

Para $((r, r^2 - n) \in S)$ {
$r^2 - n = \prod_{i=1}^{k} p_i^{e_i}$;
$\vec{v}(r^2 - n) = (e_1, e_2, \ldots e_k)$; }
Construa a matriz de ordem $(k+1) \times k$ com os vetores $\vec{v}(r^2 - n)$ reduzidos (mod 2);
Através da Eliminação Gaussiana encontre uma combinação linear de vetores $\vec{v}(r_1) + \vec{v}(r_2) + \ldots + \vec{v}(r_k) = \vec{0}$.
$x = r_1 \cdot r_2 \cdot \ldots \cdot r_k \bmod n$;
$y = \sqrt{(r^2_1 - n)(r^2_2 - n)\ldots(r^2_k - n)} \bmod n$
$fator1 = mdc(x + y, n)$;
$fator1 = mdc(x - y, n)$;
retorne fator1, fator2;
[Fim algoritmo]

3.2. Implementação do Crivo Quadrático

3.2.1. Ambiente de desenvolvimento

O método Crivo Quadrático, foi implementado na linguagem C em uma plataforma Pentium IV, 3200 MHz, 512 MB de RAM, usando o sistema operacional Linux Ubuntu. Para a representação de números grandes de tamanho arbitrário foi utilizada a biblioteca GMP, versão 4.3.1, escrita na linguagem C, disponibilizada no site http://gmplib.org/.

3.2.2. Etapas da implementação

O primeiro passo do método Crivo Quadrático é encontrar uma base de fatores. Os números pertencentes a essa base, devem satisfazer as seguintes condições:

- Ser primo
- Ser menor que um limite dado B
- Atender a condição do Símbolo de Legendre onde $(n/p)=1$

Para isso foi utilizada a função mpz_nextprime, que lista os primos até o limite B especificado. Para cada primo foi calculado o símbolo de Legendre. Assim foi criado um vetor com os números para os quais as condições foram satisfeitas.

O segundo passo é determinar um conjunto de números que podem ser completamente fatorados sobre a base de fatores. Para isso, inicialmente foram calculados os $f(r_i)$'s usando a fórmula $f(r_i)=r_i^2-n$. Na implementação desta fórmula foram utilizadas as funções de adição mpz_add_ui, potenciação mpz_pow_ui e subtração mpz_sub.

Para determinar se os números $f(r_i)$'s são completamente fatorados pela base de fatores foi utilizada a função mpz_cmp_ui, que compara os fatores $f(r_i)$'s com os primos da base de fatores. Os $f(r_i)$'s que satisfazem essa condição são armazenados em um vetor. O mesmo procedimento é feito para os r_i's correspondentes.

O terceiro passo é usar a Eliminação Gaussiana para encontrar um produto dos números determinados no 2º passo que seja um quadrado perfeito. Para isso é criada uma matriz determinada pelos $f(r_i)$'s, onde cada linha representa um vetor de 0's e 1's correspondentes a um $f(r_i)$, como segue: em primeiro lugar é calculada a fatoração do $f(r_i)$ dado; se o expoente do fator de $f(r_i)$ for par atribui-se 0 e, se for ímpar atribui-se 1.

Em seguida é utilizada a função inicializa_matriz_identidade, que cria uma matriz identidade cuja ordem é dada pelo número de $f(r_i)$'s determinados. Após essas etapas, chama-se a função eliminação_gaussiana_mod_2, que realiza a Eliminação Gaussiana nas duas matrizes, procurando uma ou mais linhas nulas. Encontrada uma linha nula, deve-se tomar a linha correspondente da matriz identidade onde os 1's que aparecerem correspondem aos r_i's determinados anteriormente. Para calcular *x* e *y* como especificado no algoritmo apresentado foi utilizada, além das funções de multiplicação e cálculo de raiz quadrada, a função de operação modular mpz_mod. O mdc foi calculado utilizando-se a função mpz_gcd. Dessa forma foram retornados os fatores de *n*.

3.2.3. Testes e análise dos resultados

Foram testados números compostos (n) contendo de 2 a 30 dígitos decimais. O tempo de execução do algoritmo foi medido em segundos. Os resultados obtidos podem ser visualizados na Figura 1 a seguir.

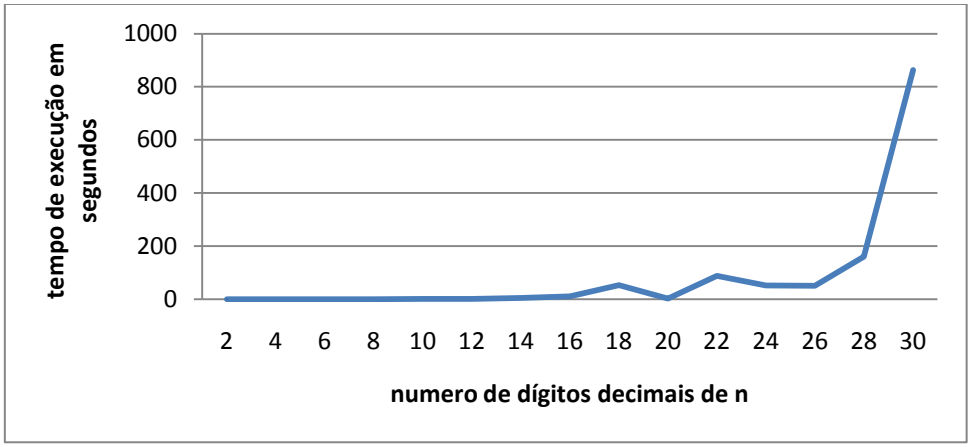

Figura 1: Tempo de execução

Com base nos resultados obtidos, percebe-se que o tempo fatoração aumenta exponencialmente de acordo com a quantidade de dígitos de n. Não foram apresentados resultados de números maiores devido ao aumento muito significativo do tempo de execução e à problemas de "overflow" (memória insuficiente). Para uma implementação mais eficiente são necessários estudos mais aprofundados de cada etapa do processo de fatoração.

5. Agradecimentos

Os autores agradecem pelo apoio financeiro (bolsa) concedido pela Universidade Estadual de Mato Grosso do Sul e também por todos os acadêmicos, coordenadores, professores e técnicos que, diretamente ou indiretamente, contribuíram para a realização desse trabalho.

6. Referências

Crandall, R., Pomerance, C. 2002. **Prime Numbers- A Computational Perspective.** New York: Springer-Verlag, 1º Edição, 227p.

Coutinho, S. 2000. **Números inteiros e Criptografia RSA**. IMPA-SBM.

Pomerance, C. 1985. **The Quadratic Sieve Factoring Algorithm**. New York: Springer-Verlag , Ed. T. Beth, N. Cot, and I. Ingemarsson , 13p.