CRIVO QUADRÁTICO: IMPLEMENTAÇÃO DA OBTENÇÃO DE UM CONJUNTO DE

NÚMEROS COMPLETAMENTE FATORADOS SOBRE UMA BASE DE FATORES

Alex Zanella Zaccaron1; Adriana Betânia de Paula Molgora2

1Estudante do Curso de Ciência da Computação da UEMS, Unidade Universitária de Dourados;

E-mail: alex_zaccaron@hotmail.com. Bolsista PIBIC/UEMS.

2Professora 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 estudo da fatoração de inteiros é extremamente importante por estar diretamente relacionado com a segurança de sistemas criptográficos, como o RSA. O Crivo Quadrático é um dos métodos de fatoração mais importantes da atualidade. Esse trabalho tem como objetivo a implementação de uma das etapas desse método de fatoração que consiste em determinar um conjunto de números que possam ser completamente fatorados sobre uma base de fatores.

Palavras-chave: Criptografia. Fatoração. Número Primos.

1. Introdução

A fatoração de números inteiros é dos problemas mais antigos da humanidade que ainda vem sendo estudado até os dias atuais por conta de sua relação com sistemas de criptografia como o RSA, e, de acordo com [2], o Crivo Quadrático é um dos métodos de fatoração mais potentes.

Pode-se dizer que a fatoração através do método Crivo Quadrático é desenvolvida em duas etapas principais. A primeira etapa consiste na obtenção de um conjunto de números que possam ser completamente fatorados sobre uma base de fatores. A segunda etapa está relacionada com a determinação de um produto de números obtidos na primeira etapa que seja um quadrado perfeito.

Este projeto tem como objetivo realizar uma implementação da primeira etapa do método Crivo Quadrático.

2. Materiais e Métodos

Para o alcance do objetivo proposto, o trabalho foi distribuído em três fases, sendo elas:

1. Estudo do manual da biblioteca GMP;
2. Integração da GMP em um compilador C;
3. Documentação e implementação do método de fatoração de inteiros Crivo Quadrático,

utilizando as funções da biblioteca GMP.

3. Resultados e Discussão

Neste capítulo é apresentado os conceitos básicos do funcionamento do Crivo Quadrático, detalhes da implementação do método em questão e os resultados dos testes realizados com o algoritmo desenvolvido.

3.1. Crivo Quadrático

De acordo com [1] e [2], 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 $x^2 \equiv y^2 (mod\ n)$, então tem-se que $(x+y)(x-y) \equiv 0 (mod\ n)$. Logo, $\frac{n}{(x^2-y^2)} = (x+y)(x-y)$ e os números $d = mdc(x+y, n)$ e $f = mdc(x-y, n)$ poderão ser fatores não triviais de n. Ou seja, a ideia básica do método consiste em encontrar congruências da forma $x_i^2 \equiv y_i (mod\ n)$, onde $\prod y_i = y^2$ é um quadrado perfeito. Se $x = \prod x_i$, então $x^2 \equiv y^2 (mod\ n)$.

Ainda segundo [1] e [2], para encontrar x e y, na prática, primeiramente é necessário determinar uma base de fatores, que é um conjunto de números primos $p_i \leq B$ onde B é um limite dado, e, cada primo p o número n deve ser um resíduo quadrático módulo p. Em seguida, a partir da função $f(x_i) = x_i^2 - n$, são calculados os chamados $f(x_i)$s para x_is próximos à $\sqrt{n}$ que serão, se possível, completamente fatorados pela base de fatores.

3.2. Implementação do Crivo Quadrático

A partir dos estudos realizados, foi desenvolvido um algoritmo cujos detalhes são apresentados a seguir.

3.2.1. Ambiente de desenvolvimento

A Primeira etapa do método Crivo Quadrático foi implementado na linguagem C em uma plataforma Pentium Dual Core, 1.73 GHz, 2 GB de RAM, utilizando o o compilador GCC e o sistema operacional Linux Ubuntu 10. Para a representação de números grandes de tamanho arbitrário foi utilizada a biblioteca GMP, versão 5.0.4, escrita na linguagem C, disponibilizada no site http://gmplib.org/.

3.2.2. Etapas da implementação

Para um melhor entendimento e facilidade na implementação, a primeira etapa do método foi dividida em dois passos:

- Obtenção da base de fatores;
- Determinação e fatoração dos $f(x)$s.

3.2.2.1. Obtenção da base de fatores

Na implementação do primeiro passo, que consiste em determinar todos os números primos até um limite B, onde para cada primo p, n deve ser resíduo quadrático módulo p, foi utilizado o crivo de Eratóstenes, estudado em [1] e [3], para determinar todos os números primos e a função *mpz_legendre* para a verificação se para o primo encontrado n é um resíduo quadrático módulo p, também dito como símbolo de Legendre igual à 1. Todos os primos

satisfeitos, que constituem a base de fatores foram armazenados em um arquivo para o segundo passo utilizá-los.

3.2.2.2. Determinação e fatoração dos *f(x)*s

Na implementação do segundo passo, foi utilizada as funções *mpz_pow_ui* e *mpz_sub* para calcular $f(x) = x^2 - n$. Depois de calculada a função, o algoritmo testa todos os primos da base de fatores, se $f(x)$ é dividido por algum primo então ocorre a divisão, então, da mesma forma, o resto da divisão é testado, assim sucessivamente. Ao final, se o resto da divisão for 1, então o $f(x)$ foi completamente fatorado pela base de fatores. O próximo $f(x)$ é calculado e o processo é refeito. A quantidade de $f(x)$ s a ser verificado é $2M + 1$ de onde M, da mesma forma que B, é um valor dado.

Todas as funções da biblioteca GMP citadas neste trabalho estão presentes em [4].

3.2.3. Testes e análise dos resultados

O algoritmo desenvolvido foi submetido a vários testes tendo como principal objetivo a verificação do tempo gasto e a quantidade de memória utilizada durante a execução.

3.2.3.1 Obtenção da base de fatores

Para cada valor de B, foram testados quatro números n: 100, 10^{30}, 10^{60} e 10^{90}. O campo tempo, apresentado na tabela 1 a seguir, é a média entre os tempos de execução para os números distintos. A variação do tempo de execução do menor para o maior número é de aproximadamente 3.06%, ou seja, o tempo apresentado na tabela possui uma margem de variação de 1.53% para mais ou para menos.

A memória utilizada é equivalente à B bytes. Isso ocorre por conta de que o Crivo de Eratóstenes constrói um vetor de B posições onde cada posição utiliza 1 byte de memória. O tamanho da base de fatores |BF|, não varia, nesse caso, em relação aos valores diferentes de n.

Valor de *B*	**Tempo (s)**	**Memória**	**|BF|**
100	0	88 kB	23
100000	0.005	180 kB	9590
1000000	0.065	1 MB	78496
10000000	0.875	9.6 MB	664577
100000000	9.35	95.5 MB	5761453
1000000000	96.91	953.8 MB	50847532

Tabela 1*: Teste dos valores de B e n*

3.2.3.2 Determinação e fatoração dos *f(x)*s

Neste passo foram realizados testes com diferentes valores de n: 100, 10^{30}, 10^{60} e 10^{90}, para cada n, foi verificado o tempo de execução para os valores de M: 100, 1000 e 1000, apresentados na tabela 2.

Pode-se observar que o tempo de execução para os valores com 31, 61 e 91 dígitos não teve uma variação muito significativa. A partir dos resultados apresentados na tabela 2,

pode-se concluir que o tempo de execução para este passo não sofre uma influência significativa para valores de n entre 30 e 90 dígitos.

n	**|BF|**	***M***	***f(x)*s fatorados**	**Tempo (s)**
100	1227	100	80	0.01
		1000	800	0.07
		10000	7998	1.85
10^{30}	1227	100	0	0.08
		1000	7	0.8
		10000	36	7.95
10^{60}	1227	100	0	0.09
		1000	0	0.82
		10000	0	8.5
10^{90}	1227	100	0	0.09
		1000	0	0.84
		10000	0	8.5

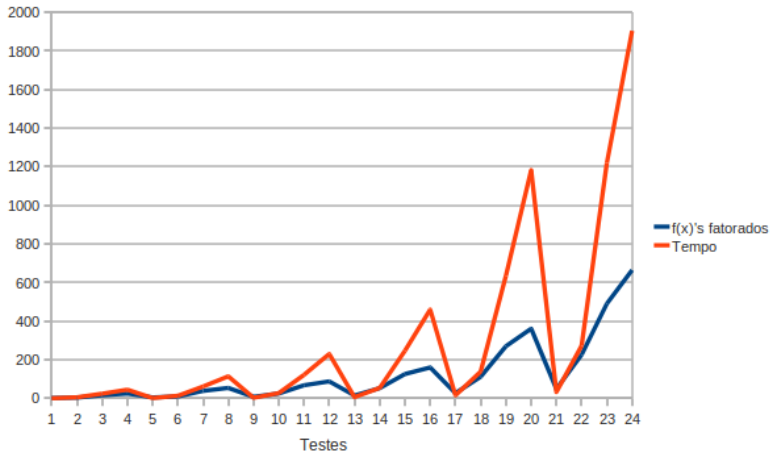
Tabela 2: Tempo de fatoração dos f(x)s com diferentes valores de n.

3.2.3.3 Testes

Nesta seção são apresentados os resultados das baterias de testes realizadas com o algoritmo desenvolvido. As três sequências de testes demonstram o tempo necessário para a execução da primeira etapa do Crivo Quadrático para um dado valor N.

3.2.3.3.1 Bateria de testes 1

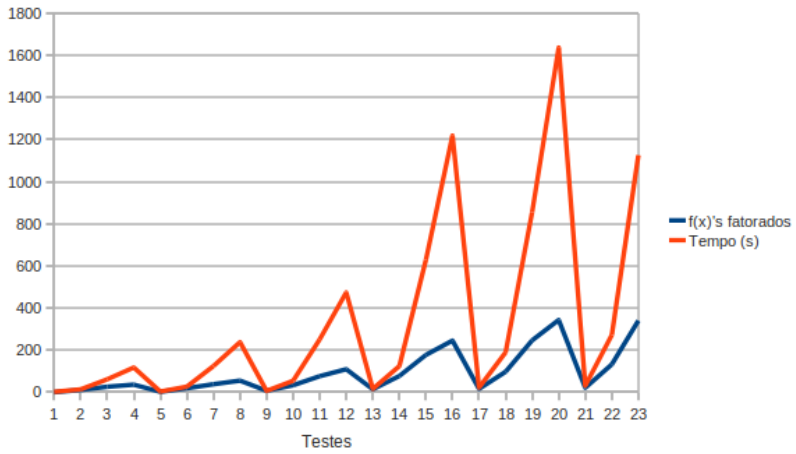
O gráfico 1 apresenta os resultados da primeira bateria de testes utilizando o seguinte número $N = 746381928376542312984756286547$, composto por 30 casas decimais.



Analisando o gráfico acima, o 22^{o} teste teve um bom resultado, nesse teste 20001 $f(x)$s foram verificados, 226 $f(x)$s foram completamente fatorados por uma base de fatores contendo 39192 primos em 273 segundos.

3.2.3.3.1 Bateria de testes 2

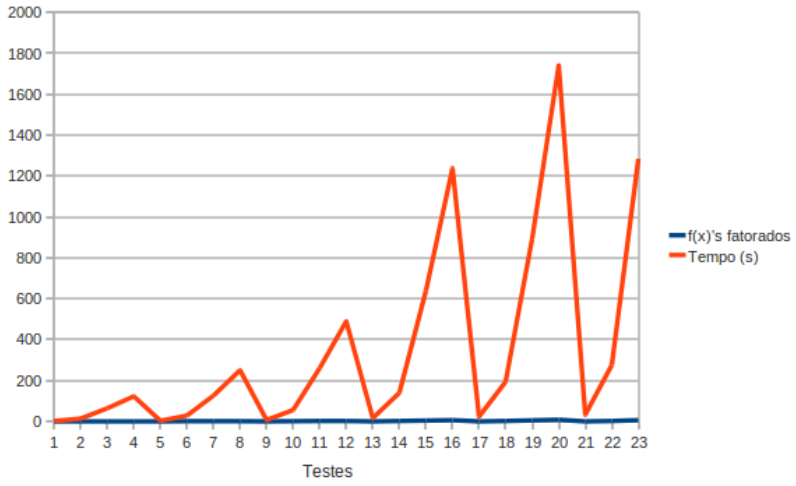
O gráfico a seguir mostra os resultados adquiridos na segunda bateria de testes, realizado com o seguinte número $N = 263049224041388331877713429325367151752 9$, composto por 40 casas decimais e sendo o produto dos seguintes números primos: 481129598370820486 e 54673257461630679457.



Nessa bateria de testes houve um aumento no tempo de execução em relação a quantidade de $f(x)$s fatorados, pois N é maior que o anterior. O 22^{o} teste obteve um dos melhores resultados dessa bateria de testes, onde 132 $f(x)$s foram completamente fatorados por uma base de fatores contendo 39128 primos, em 271 segundos.

3.2.3.3.1 Bateria de testes 3

O gráfico a seguir apresenta o resultado da terceira bateria de testes usando o número $N = 189620700613125325959116839007395234454467716598457179234 02$, composto por 60 dígitos e sendo a multiplicação entre os números primos $671998030559713968361666935769$ e $282174488599599500573849980909$.



Analisando o gráfico acima, podemos perceber que a quantidade de $f(x)$s fatorados é muito baixa, sendo que nenhum teste obteve um bom resultado. Para tal objetivo, seria necessário aumentar o valor de M e/ou B, porém, o tempo necessário para a execução com valores altos torna-se inviável para este trabalho, pois ultrapassaria trinta minutos de execução.

4. Conclusão

O estudo de métodos de fatoração, como o Crivo Quadrático, possui grande importância, pois está totalmente ligado à alguns sistemas de criptografia de chave pública, como o RSA.

Este trabalho apresentou diversos testes com a implementação da primeira etapa do Crivo Quadrático, onde se pode perceber que em ambas as baterias de testes 1 e 2, o 22^{o} teste obteve um dos melhores resultados. Nesse teste verificou-se que, para números de 30 e 40 dígitos, os possíveis valores ideais para B e M são $B \approx 10000$ e $M \approx \frac{B}{10}$. A terceira bateria de testes não pode ser levada em consideração, pois não obteve nenhum resultado aceitável, por conta de que o tempo de fatoração é muito alto.

Para trabalhos futuros sugere-se novos testes com o algoritmo e um estudo mais aprofundado do método em questão, a fim de melhorar a implementação do algoritmo para reduzir o tempo de execução.

5. Agradecimentos

Gostaria de agradecer à PIBIC/UEMS pela oportunidade e concessão da bolsa, e à Professora Adriana Molgora pela paciência e incentivo durante a orientação.

6. Referências

Livros

[1] CRANDALL, R., POMERANCE, C. **Prime Numbers- A Computational Perspective**. 2 Ed. New York: Springer-Verlag, 2005. 597 p.

[2] POMERANCE, C. **The Quadratic Sieve Factoring Algorithm**. New York: Springer - Verlag , 1985. p. 169– 182 .

[3] COUTINHO, S.C. **Números Inteiros e Criptografia RSA**. Rio de Janeiro: IMPA, 2011. 226 p.

Manuais

[4] GRANLUND, T. **GNU multiple precision arithmetic library 5.0.4**. Disponível em:

http://gmplib.org. Acesso em: 30 de junho 2012.