

# Disciplina de Análise e Síntese de Algoritmos - 2003/2004

## Professor Arlindo Oliveira

### Algoritmos para factorização de números inteiros

Vitor Miguel Neves Fernandes (5490/M)  
MEIC, DEI, IST  
Av. Rovisco Pais, 1049-001 Lisboa  
vmnf@mega.ist.utl.pt  
30 de Janeiro de 2004

---

#### Sumário

*A factorização de números inteiros é um desafio desde os tempos mais antigos. Nos anos setenta dois investigadores factorizaram o número  $F_7 = 2^{2^7} + 1$ . Depois nasceu a criptografia com o algoritmo RSA, baseada em números primos, cuja chave factorizada permite descodificar a maior parte das mensagens encriptadas actualmente. Em 1999 uma equipa de cientistas de seis países diferentes factorizaram um número com 512bit que representa 95% das comunicações seguras sobre a Internet. O relatório que se apresenta analisa a evolução do processo de factorização de números inteiros e alguns dos seus algoritmos.*

#### Palavras-chave

*Algoritmos, factorização, números primos, complexidade, RSA, encriptação e segurança.*

---

## 1. INTRODUÇÃO

Em matemática o problema de factorização de números inteiros é obter, a partir de um inteiro positivo, obter a sua decomposição em números primos. Por exemplo, para o número 36 a sua decomposição é  $2^2 \cdot 3^2$ . Esta factorização é única de acordo com o teorema fundamental da matemática. Este problema é de interesse para um conjunto de áreas como sejam a da Matemática, Criptografia, Teoria da Complexidade e Computadores Quânticos.

Dados dois números primos grandes é fácil multiplicá-los. No entanto, dado o resultado do produto, é complicado obter os dois factores. Esta é a base do sistema de encriptação de dados RSA (elaborado por Rivest, Shamir e Adleman) e do gerador de números aleatório Blum Blum Shub. Caso seja obtido um algoritmo que realize a factorização rapidamente, os sistemas criptográficos baseados em factorização serão desaconselhados ou mesmo inúteis.

## 2. FACTORIZAÇÃO DE NÚMEROS INTEIROS

### 2.1 Os Números Primos

Um número primo é um número inteiro  $p > 1$  que não tem divisores inteiros a não ser o número 1 e  $p$ . Por exemplo os únicos divisores do número 13 são o número 1 e o número 13, enquanto o número 24 tem por divisores os números 1, 2, 3, 4, 6, 8, 12 e 24 (a que corresponde a factorização em números primos  $24 = 2^3 \cdot 3$ ), logo este número não é primo. O 1 não é considerado primo nem número composto (Wells86)

embora existam autores que considerem primo. Tendo então excluído o número 1 temos que o menor número primo é o 2. Os primeiros números primos são o 2, 3, 5, 7, 11, 13, 17, 19, 37,... À medida que avançamos para números primos maiores podemos levantar a questão se “Após determinado número primo todos os restantes serão compostos?” e a resposta é não (demonstrado por Euclides). O maior número primo conhecido em 2003 é o primo Mersenne  $2^{20996011} - 1$  (Weisstein03). De especial especial interesse temos também os números primos de Fermat que apesar das suas expectativas apenas 5 são conhecidos sendo o maior o 65537. Os números primos de Fermat permitem obter um polígono regular usando apenas usando uma régua e um compasso (descoberta efectuada por Gauss com apenas 19 anos).

**2.2 Quantos Primos Existem?**

Se supusermos que o número de primos é finito, sendo o  $p_r$ , vamos multiplicar todos os primos e adicionar 1:  $n = p_1 \cdot p_2 \cdot \Lambda \cdot p_r + 1$  temos então que  $n > p_r$  e logo não pode ser primo pois assumiu-se  $p_r$  como sendo o maior. O número  $n$  terá então de ser composto. A forma como foi construído impede que um dos seus factores seja um dos primos utilizados, pois se utilizarmos todos estes para dividir  $n$  obtemos 1 como resto da divisão. Isto implica que terá de haver um número primo maior que  $p_r$  e que divida  $n$ , o que é uma contradição!

A fórmula seguinte de um modo geral permite obter um número primo  $2+1=3$ ,  $2.3+1=7$ ,  $2.3.5+1=31$ ,  $2.3.5.7+1=211$ , ... que são todos primos, no entanto  $2.3.5.7.11.13+1=30031=59.509!$

**2.3 O crivo de Eratosthenes**

A obtenção de números primos não se consegue à custa de fórmulas mas através de testes de primalidade ou por crivos como o criado por Eratosthenes (276 AC a 196 AC) na Grécia clássica em que estes são encontrados por tabelas como a Fig. 1 (à direita).

O crivo é construído numa tabela com 6 colunas em que após a primeira linha os números primos apenas surgem na primeira e na quinta coluna (eliminação dos números divisíveis por 2 e 3). Para eliminar os números divisíveis por 5 e 7 temos de eliminar os números que estão em algumas diagonais a  $45^\circ$ , os números divisíveis por 11 e 13 aparecem numa deslocação igual à de um cavalo num tabuleiro de xadrez e com este processo relativamente simples eliminaram-se todos os números compostos até  $17.19 = 323$ .

No presente existem crivos muito sofisticados usados para encontrar números primos muito grandes para usar por exemplo em chaves públicas (ou para as descriptar!).

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42
43	44	45	46	47	48
49	50	51	52	53	54
55	56	57	58	59	60
61	62	63	64	65	66
67	68	69	70	71	72
73	74	75	76	77	78
79	80	81	82	83	84
85	86	87	88	89	90
91	92	93	94	95	96
97	98	99	100		

**Fig. 1: O crivo de Eratosthenes (módulo 6)**

**2.4 Outros Métodos para obter Números Primos**

**2.4.1 Um teorema chinês**

Trata-se de um teste de primalidade (permite verificar se dados número é ou não primo) que diz que um número  $n$  é primo se e só se este divide  $2^n - 2$ . Este teste é válido apenas para números tal que  $n < 341$ , por exemplo para  $n = 5$ ,  $2^n - 2 = 30$  que é divisível por 5,  $n = 6$ ,  $2^n - 2 = 62$  que não é divisível por 6 (mas 6 também não é primo!). Ao chegar a  $n = 341$ , o teste falha pois  $(2^{341} - 2)/341$  cujo resto da divisão é zero e no entanto 341 é composto  $341 = 11.31$ .

**2.4.2 Primos de Mersenne**

São números primos obtidos pela fórmula  $M_p = 2^p - 1$ , onde p é primo. Caso p fosse composto considerando  $n = pq$  e  $2^n - 1 = (2^p - 1)(2^{p(q-1)} + 2^{p(q-2)} + \Lambda + 1)$  que verifica que origina também um número composto. Em 18 de Novembro 1978 o record de detecção de números primos de Mersenne estava em  $2^{11213} - 1$  e em 2003 em  $2^{20996011} - 1!$

### 2.5 A Importância dos Números Primos

A importância é de tal ordem que alguns números têm até valor comercial (em algoritmos de encriptação como seja a encriptação de chave pública RSA) de tal modo que R. Schlafly em 1994 obteve uma patente nos Estados Unidos da América (U.S. Patent 5,373,560) para os dois números primos que se apresentam de seguida.

```
98A3DF52AEAE9799325CB258D767EBD1F4630E9B
9E21732A4AFB1624BA6DF911466AD8DA960586F4
A0D5E3C36AF099660BDDC1577E54A9F402334433
ACB14BCB
```

```
93E8965DAFD9DFEFCFD00B466B68F90EA68AF5DC9
FED915278D1B3A137471E65596C37FED0C7829FF
8F8331F81A2700438ECDCC09447DC397C685F397
294F722BCC484AEDF28BED25A AAB35D35A65DB1F
D62C9D7BA55844FEB1F9401E671340933EE43C54
E4DC459400D7AD61248B83A2624835B31FFF2D95
95A5B90B276E44F9.
```

**Fig. 2: Números primos registados por R. Schlafly em 1994 (base hexadecimal)**

### 2.6 Factorização de Números

A factorização de números é o processo que permite decompor qualquer número nos seus factores primos constituintes. Para qualquer inteiro positivo  $n \geq 2$  é possível determinar  $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_k^{\alpha_k}$  onde  $p_i$  são os k factores primos com expoente  $\alpha_i$  (ver Fig. 3).

Em geral, a factorização em números primos é um problema complicado e foram desenvolvidos algoritmos sofisticados para tratar números que se enquadram em casos especiais.

Os números inteiros também podem ser factorizados com primos de Gauss como se apresenta na Fig. 4.

Uma abordagem possível para determinar a factorização de um número inteiro é testar se o resultado da divisão por uma sequência crescente de números apresenta resto zero. Se tal suceder está encontrado um dos factores do número em questão (Factorização por Pesquisa Directa).

Esta abordagem realiza  $\lfloor \sqrt{n} \rfloor$  operações de divisão e é um dos processos mais lentos de determinação dos factores de um número.

#### 2.6.1 Factos históricos

Em 1994 números com 512bit (155 dígitos) pensava-se que seria praticamente impossível de obter a sua factorização. No entanto realizaram-se desenvolvimentos importantes em áreas como a da matemática e a da computação e neste momento é difícil prever os próximos desenvolvimentos. Quando em 1977 Rivest desafiou o mundo para factorizar a RSA-129, um número com 129 dígitos, estimou que com métodos tradicionais levaria aproximadamente  $10^{16}$  anos de tempo de computação.

Dezassete anos depois (em 1994), num esforço cooperativo que demorou “somente” 8 meses de computação para factorizar o número apresentado. Convém não esquecer que resta a possibilidade de inventar novos métodos computacionais que transformem a factorização num processo “simples”.

Em Abril de 1999 factorizou-se um número com 211 dígitos -  $(10^{211} - 1)/9$  - comparável ao esforço de factorizar a RSA-140.

A RSA-155 requereu um total de 35 anos de tempo de computação em 300 máquinas (SGI, SUN Workstations e PCs Pentium II). Como o trabalho foi realizado principalmente em paralelo este foi concluído em somente 7 meses.

n	fact.	n	fact.
1	1	11	11
2	2	12	$2^2 \cdot 3$
3	3	13	13
4	$2^2$	14	$2 \cdot 7$
5	5	15	$3 \cdot 5$
6	$2 \cdot 3$	16	$2^4$
7	7	17	17
8	$2^3$	18	$2 \cdot 3^2$
9	$3^2$	19	19
10	$2 \cdot 5$	20	$2^2 \cdot 5$

**Fig. 3: Factorização em números primos**

n	factorização
1	1
2	$-i(i+1)^2$
3	3
4	$-(i+1)^4$
5	$-i(2i+1)(2+i)$
6	$-3i(i+1)^2$
7	7
8	$i(i+1)^6$
9	$3^2$
10	$-(i+1)^2(2i+1)(i+2)$

**Fig. 4: Factorização em primos gaussianos**

No presente o algoritmo mais rápido para factorização de números inteiros são baseados no Crivo de Campo Numérico (Number Field Sieve ou NFS). Os seis maiores números factorizados com o NFS têm 174 (em 3 de Dezembro de 2003), 160 (em 1 de Abril de 2003), 158 (em 19 de Janeiro de 2002), 155, 140 e 130 dígitos decimais. Existe também uma variante o Special Number Field Sieve (SNFS) mais rápida mas que apenas pode ser usada em alguns casos especiais. Esta variante já factorizou números com 244 (em 3 de Janeiro de 2003), 227 (em 22 de Janeiro de 2002), 233 (em 14 de Novembro de 2000), 211 (em 8 de Abril de 1999) e 186 (em Setembro de 1998).

O processamento paralelo combinado de supercomputadores cria-nos a expectativa de conseguir factorizar números de 512bit em somente alguns dias.

### 2.7 Qual Classe de Complexidade da Factorização de Números Inteiros

Não é sabido ao certo qual a classe de complexidade dos algoritmos para factorização de números inteiros. O problema de decisão “N tem um factor menor do que M?” é reconhecido como estando em NP e co-NP. isto resulta do facto de as respostas SIM ou NÃO poderem ser verificadas desde que sejam dados os factores primos em conjunto com os seus testes de primalidade. Suspeita-se que o problema esteja fora das três classes de complexidades P, NP-Completo e co-NP-Completo. Se se conseguisse provar que estava em NP-Completo ou co-NP-Completo isso implicaria que NP=co-NP. Isto seria um resultado surpreendente. Por enquanto suspeitasse que esteja fora de ambas as classes. Muitos investigadores já tentaram encontrar algoritmos que resolvam o problema em tempo polinomial mas até agora falharam e por isso pensasse que também esteja fora de P.

## 3. ALGORITMOS PARA FACTORIZAR NÚMEROS INTEIROS

Vários algoritmos foram criados com o objectivo de factorizar números inteiros. Os algoritmos para factorização variam em sofisticação e complexidade

É muito difícil criar um algoritmo genérico que sirva o propósito eficiência e abrangência. De um modo geral existem algoritmos abrangentes, mas que pela sua velocidade, para números grandes, são impraticáveis enquanto que os algoritmos eficientes são destinados apenas a parcelas do universo de problemas. Características do número a factorizar ou de alguns dos seus factores podem poupar muito tempo no processo de pesquisa da solução

O método mais simples é o método da Pesquisa Directa também conhecido por Divisão por Tentativas que testam se conseguem dividir o número apresentado. Este algoritmo é usado apenas para números pequenos.

O algoritmo determinista mais rápido conhecido é o Pollard-Strassen (Pomerance87, Hardy90).

### 3.1 A Complexidade

A complexidade dos algoritmos para determinar a factorização de números inteiros varia de forma elevada (ver Fig. 5), por exemplo para um número com 20 dígitos o número de operações pode variar entre  $\cong 10.000$  e  $\cong 10.000.000$  (Fig. 6)!

O algoritmo mais eficiente para números grandes (com mais de 28 dígitos) é o Numeric Field Sieve ou Crivo por Campo Numérico.

Método	Complexidade
Crivo simples (Sieve)	$p$
Pollard rho - $\rho$	$(p)^{1/2}$
Curvas Elípticas	$L_p[1, 1/2]$
Crivo Quadrático (Quadratic Sieve - QS)	$L_n[1/2, c]$
Crivo por Campo Numérico (Numeric Field Sieve - NFS)	$L_n[1/3, c]$

**Fig. 5: Complexidade dos métodos de factorização (n é o número a factorizar, p o menor factor primo, c é uma constante que depende do algoritmo e da sua implementação e  $L_x[\alpha, c] = e^{c \ln^\alpha x (\ln \ln x)^{1-\alpha}}$ )**

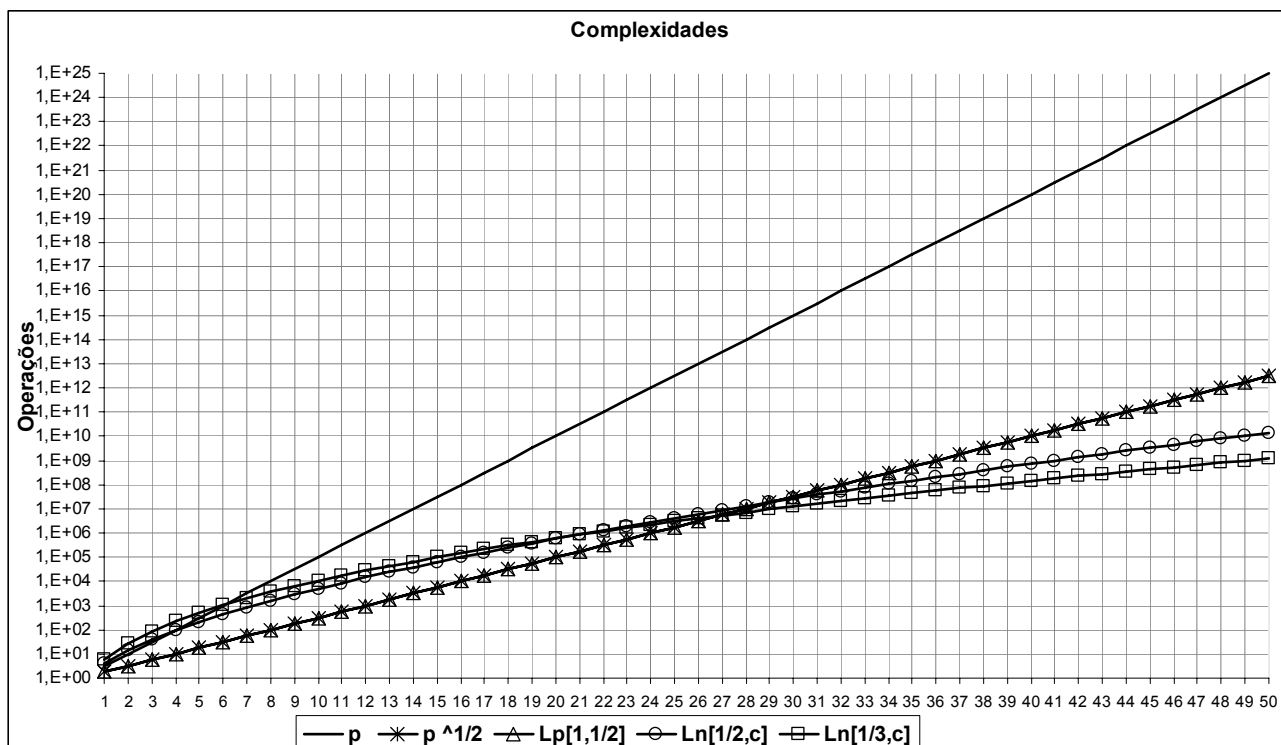


Fig. 6: Complexidade dos métodos de factorização (com  $c=1$  para  $L_n[1/2, c]$  e  $c=1,526285$  (Weisstein#13) em casos especiais para  $L_n[1/3, c]$ )

### 3.2 Os Algoritmos

#### 3.2.1 Algoritmo Pesquisa Directa

É um dos algoritmos mais simples para factorizar números inteiros e consiste em realizar tentativas de divisão sucessivas até encontrar um factor. Os múltiplos de factores pequenos são normalmente excluídos, mas o simples facto de os excluir pode tornar ainda o processo mais lento do que simplesmente considerá-los. É um algoritmo ineficiente e apenas deve ser usado para números pequenos. Para detectar todos os factores de um número é necessário realizar no máximo  $\lfloor \sqrt{n} \rfloor$  divisões. Caso o menor factor primo encontrado seja  $> \sqrt[3]{n}$  então o seu cofactor é também primo. A sua complexidade é da ordem de  $\lfloor \sqrt{n} \rfloor$  ou  $p$  (sendo  $p$  o menor factor).

#### 3.2.2 Algoritmo de Fermat

Dado um número  $n$  este método pesquisa por dois inteiros que satisfaçam a expressão  $n = x^2 - y^2$ . Como  $n = (x - y)(x + y)$  e  $n$  é factorizável. Fazendo  $a = x + y$  e  $b = x - y$  é possível chegar à expressão  $x^2 - y^2 = \frac{1}{4} [(a + b)^2 - (a - b)^2] = ab$ . O processo baseia-se no facto de apenas existirem 22 possibilidades de quadrados perfeitos em cada 100 números logo eliminamos bastantes possibilidades. Maurice Kraitchik acelerou o processo testando antes  $x^2 \equiv y^2 \pmod{n}$ .

#### 3.2.3 Algoritmo da Diferença dos Quadrados

Este método foi usado por Fermat e melhorado por Gauss. Gauss pesquisa por dois inteiros que satisfaçam a expressão  $y^2 \equiv x^2 - n \pmod{e}$  para vários  $e$ . Este método permite a exclusão de muitos potenciais factores e funciona melhor para factores próximos.

#### 3.2.4 Algoritmo de Euler

É um algoritmo que expressa  $n$  de duas formas quadráticas diferentes. Sendo  $n = a^2 + b^2 = c^2 + d^2$  então  $a^2 - c^2 = d^2 - b^2$  logo  $(a - c)(a + c) = (d - b)(d + b)$ . Se  $k$  for o GDC (maior divisor comum) de  $a - c$

e d-b então temos que  $a - c = kl$ ,  $d - b = km$  e  $(l, m) = 1$ . Continuando  $l(a + c) = m(d + b)$ , mas como  $(l, m) = 1$ ,  $m|a + c$  e  $(a + c) = mn$  temos  $b + d = l n$ .

### 3.2.5 Algoritmo Pollard rho - $\rho$

O algoritmo Pollard rho é também conhecido como Pollard Monte Carlo. O algoritmo baseia-se numa fórmula iterada até esta entrar num ciclo. A fórmula é  $x_{n+1} = x_n + a \pmod{n}$  que para qualquer valor inicial de  $x_0$  produzirá um ciclo. O tempo de espera é proporcional a  $\sqrt{n}$ . Tendo em conta que  $n = pq$  e de acordo com o Teorema Chinês do Resto é possível obter então os factores p e q.

A segunda parte deste método trata da detecção da entrada em ciclo e usa a ideia atribuída a Floyd para comparar  $x_i$  a  $x_{2i}$  para todo o i. A sua complexidade é da ordem de  $\lfloor \sqrt[4]{n} \rfloor$  ou  $\lfloor \sqrt{p} \rfloor$  (sendo p o menor factor).

#### 3.2.5.1 Algoritmo de Brent (Brent's Factorization Algorithm)

Este algoritmo melhora a segunda fase do algoritmo Pollard rho que trata da detecção de sequências periódicas. O algoritmo usa apenas o valor de  $x_i$  e se i for potência ou 2, fazer  $y = x_i$ , e a cada passo do algoritmo comparar o valor de  $x_i$  com o valor presente em y. No caso da factorização em vez de comparar  $x_i$  com y pode-se calcular  $GCD(|x_i - y|, n)$  (GDC calcula o maior divisor comum entre os argumentos). Posteriormente Brent (1980) considerou outros valores diferentes de 2, mas conclui que 2 está muito próximo do valor óptimo.

### 3.2.6 Algoritmo Pollard p-1

É um algoritmo destinado a casos especiais pode funcionar em 1 passo ou em dois passos.

A versão de um passo descobre números primos p se p-1 é um produto de primos pequenos encontrando um m tal que  $m = c^q \pmod{n}$ , onde  $p-1|q$ , com q um número grande e  $(c, n) = 1$ . Como  $p-1|q$ ,  $m \equiv 1 \pmod{p}$  então  $p|n-1$ . Existe então uma probabilidade elevada de  $n|m-1$ , e nesse caso  $GDC(m-1, n)$  será um factor de n.

A versão de dois passos é destinada para a situação em que p-1 resulta do produto de um conjunto de pequenos primos e um único primo grande.

### 3.2.7 Algoritmo Williams p+1

É uma variante do algoritmo Pollard p-1 que usa séries de Lucas para rapidamente factorizar se existir um factor p em n tal que p+1 tenha uma decomposição em números primos pequenos.

### 3.2.8 Algoritmo das Curvas Elípticas

Este algoritmo calcula um múltiplo grande de um ponto aleatório numa curva elíptica módulo número que se pretende factorizar. Tem tendência a ser mais rápido do que o método Pollard rho -  $\rho$  ou Pollard p-1. O maior factor encontrado utilizando este algoritmo foi um factor primo com 54 dígitos de um co-factor, com 127 dígitos, C tal que  $n = b^4 - b^2 + 1 = 13 \cdot 733 \cdot 7177 \cdot C$  onde  $b = 63^{43} - 1$ .

### 3.2.9 Algoritmo da Curva Elíptica de Lenstra

Outro método baseado em curvas elípticas.

### 3.2.10 Algoritmo das Frações Continuadas

O algoritmo usa os resíduos da divisão continuada de  $\sqrt{mN}$  para um determinado m de forma a obter o quadrado de um número. O algoritmo resolve a equação  $x^2 \equiv y^2 \pmod{n}$  localizando um m para o qual  $m^2 \pmod{n}$  tem o menor limite superior. A sua complexidade é da ordem  $e^{\sqrt{2 \cdot \ln n \cdot \ln \ln n}}$  e era o algoritmo mais rápido para factorização de número primos até ter surgido o Crivo Quadrático que eliminou o 2 que se encontra no interior da raiz quadrada.

### 3.2.11 Algoritmo do Crivo Quadrático

O crivo quadrático pode ser usado em conjunto com o método de Dixon para factorizar números grandes  $n$ . Este escolhe valores de  $r$  tal que  $r = \lfloor \sqrt{n} \rfloor + k$ , onde  $k=1, 2, 3, \dots$ . O algoritmo procura factores  $p$  em que  $n = r^2 \pmod{p}$ . O conjunto de valores obtido é chamado de base de factores. De seguida as congruências  $x^2 \equiv n \pmod{p}$  têm de ser resolvidas para cada valor da base. Finalmente é aplicado um crivo para encontrar os valores de  $f(r) = r^2 - n$  que podem ser totalmente factorizados com a base de factores. A eliminação de Gauss é também usada para encontrar os produtos de  $f(r)$ 's que são quadrados perfeitos. A sua complexidade é da ordem  $e^{\sqrt{\ln n \cdot \ln \ln n}}$ .

Pode também ser usado um crivo quadrático com base na parábola  $x = y^2$ . Considerando os pontos tais que  $(y^2, y)$ , para  $y=2, 3, 4, \dots$  e unindo os pontos de ambos os ramos da parábola verifica-se que os pontos que são intersectados pelos segmentos são números compostos sendo os restantes os números primos, como pode verificar na Fig.

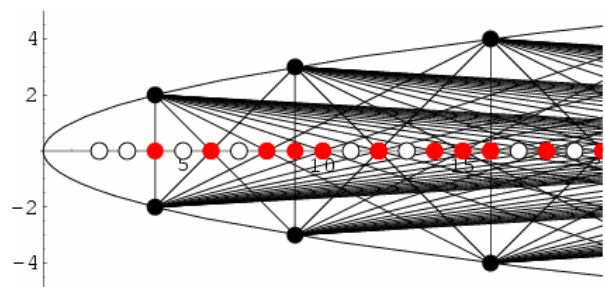


Fig. 7: Crivo quadrático

### 3.2.12 Algoritmo de Legendre

O processo é semelhante ao da Pesquisa Directa no entanto apenas se realizam as divisões para os primos de um crivo quadrático.

### 3.2.13 Algoritmo do Crivo de Campo Numérico

Trata-se de um algoritmo extraordinariamente rápido, desenvolvido por Pollard e usado para factorizar a RSA-130. Este método é o mais poderoso conhecido e apresenta uma complexidade de  $e^{c\sqrt{\ln n} \sqrt[3]{(\ln \ln n)^2}}$ . A redução do expoente foi conseguida à custa da utilização do método das Divisões Sucessivas e Crivo Quadrático. Para a constante  $c$  existem 3 valores relevantes:

1.  $c = \sqrt[3]{\left(\frac{32}{9}\right)} = 1,526 \dots$  aplicado a números perto de uma potência elevada;
2.  $c = \sqrt[3]{\left(\frac{64}{9}\right)} = 1,923 \dots$  aplicado genericamente;
3.  $c = \frac{1}{2} \sqrt[3]{92 + 26\sqrt{13}} = 1,902$  usado por Coppersmith

## 3.3 Resumo das Complexidades

Algoritmo	Em $n$	Em $p$
Pesquisa Directa	$\lfloor \sqrt{n} \rfloor$	$p$
Pollard rho - $\rho$	$\lfloor \sqrt[4]{n} \rfloor$	$\lfloor \sqrt{p} \rfloor$
Fracções Contínuadas	$e^{\sqrt{2 \cdot \ln n \cdot \ln \ln n}}$	$e^{\sqrt{2 \cdot \ln \sqrt{p} \cdot \ln \ln \sqrt{p}}}$
Crivo Quadrático	$e^{\sqrt{\ln n \cdot \ln \ln n}}$	$e^{\sqrt{\ln \sqrt{p} \cdot \ln \ln \sqrt{p}}}$
Crivo de Campo Numérico	$e^{c\sqrt[3]{\ln n} \sqrt[3]{(\ln \ln n)^2}}$	$e^{c\sqrt[3]{\ln \sqrt{p}} \sqrt[3]{(\ln \ln \sqrt{p})^2}}$

## 4. CONCLUSÕES

Para cada algoritmo apresentado, o progresso segue a Lei de Moore que diz que a velocidade dos computadores duplica em cada 18 meses. As complexidades dos algoritmos de factorização têm-se reduzido

ao longo do tempo. A álgebra linear continua a ser uma das limitações neste tipo de algoritmos. De um modo geral podemos pensar que o problema da factorização ainda está longe de ser resolvido de forma rápida o que garante alguma segurança nas comunicações com chave RSA desde que esta seja suficientemente grande.

## 5. REFERÊNCIAS

- [Cormen01] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L., “Introduction to Algorithms”, 2<sup>nd</sup> Edition, pp. 887-901, 2001.
- [Hardy90] Hardy, K.; Muskat, J. B.; and Williams, K. S. "A Deterministic Algorithm for Solving in Coprime Integers  $u$  and  $v$ ." Math. Comput. pp. 55, 327-343, 1990.
- [Pomerance87] Pomerance, C. "Fast, Rigorous Factorization and Discrete Logarithm Algorithms." In Discrete Algorithms and Complexity (Ed. D. S. Johnson, T. Nishizeki, A. Nozaki, and H. S. Wilf). New York: Academic Press, pp. 119-143, 1987.
- [Schroeder] Schroeder, M. R., “Number Theory in Science and Communication”, pp. 1-39
- [Wells86] Wells, D. The Penguin Dictionary of Curious and Interesting Numbers. Middlesex, England: Penguin Books, pp. 31, 1986.
- [Weisstein03] Weisstein, E. W. "40th Mersenne Announced." MathWorld Headline News, Dec. 2, 2003. <http://mathworld.wolfram.com/news/2003-12-02/mersenne/>.
- [Weisstein# \*] Eric W. Weisstein. “Brents Factorization Method”, “Class Group Factorization Method”, “Continued Fraction Factorization Algorithm”, “Direct Search Factorization”, “Dixon's Factorization Method”, “Elliptic Curve Factorization Method”, “Euler's Factorization Method”, “Excludent Factorization Method”, “Fermat's Factorization Method”, “Legendre's Factorization Method”, “Lenstra Elliptic Curve Method”, “Number Field Sieve”, “Pollard p-1 Factorization Method”, “Pollard Rho Factorization Method”, “Prime Factorization”, “Prime Number”, “Quadratic Sieve”, “Quiteprime”, “Trial Division”, “Veryprime”, “Williams p Plus 1 Factorization Method.” From MathWorld--A Wolfram Web Resource.  
 #01 <http://mathworld.wolfram.com/PrimeFactorization.html>  
 #02 <http://mathworld.wolfram.com/PrimeFactorizationAlgorithms.html>  
 #03 <http://mathworld.wolfram.com/BrentsFactorizationMethod.html>  
 #04 <http://mathworld.wolfram.com/ContinuedFractionFactorizationAlgorithm.html>  
 #05 <http://mathworld.wolfram.com/DirectSearchFactorization.html>  
 #06 <http://mathworld.wolfram.com/DixonsFactorizationMethod.html>  
 #07 <http://mathworld.wolfram.com/EllipticCurveFactorizationMethod.html>  
 #08 <http://mathworld.wolfram.com/EulersFactorizationMethod.html>  
 #09 <http://mathworld.wolfram.com/ExcludentFactorizationMethod.html>  
 #10 <http://mathworld.wolfram.com/FermatsFactorizationMethod.html>  
 #11 <http://mathworld.wolfram.com/LegendresFactorizationMethod.html>  
 #12 <http://mathworld.wolfram.com/LenstraEllipticCurveMethod.html>  
 #13 <http://mathworld.wolfram.com/NumberFieldSieve.html>  
 #14 <http://mathworld.wolfram.com/Pollardp-1FactorizationMethod.html>  
 #15 <http://mathworld.wolfram.com/PollardRhoFactorizationMethod.html>  
 #16 <http://mathworld.wolfram.com/PrimeNumber.html>  
 #17 <http://mathworld.wolfram.com/QuadraticSieve.html>  
 #18 <http://mathworld.wolfram.com/Quiteprime.html>  
 #19 <http://mathworld.wolfram.com/TrialDivision.html>  
 #20 <http://mathworld.wolfram.com/Veryprime.html>

#21 <http://mathworld.wolfram.com/WilliamsPlus1FactorizationMethod.html>  
 #22 <http://mathworld.wolfram.com/news/2002-08-07/primetest/http://mathworld.wolfram.com/NumberFieldSieve.html>  
 © 1999-2004 Wolfram Research, Inc.

[Internet01] ERCIM News No.39 - October 1999, “Security of E-commerce threatened by 512-bit Number Factorization”, [http://www.ercim.org/publication/Ercim\\_News/enw39/512.html](http://www.ercim.org/publication/Ercim_News/enw39/512.html)

[Internet02] “Computational number theory and data security”,  
<http://db.cwi.nl/projecten/project.php4?prjnr=84>

[Internet03] “Why the HardEncrypt Package is uncrackable”,  
<http://hcsoftware.sourceforge.net/HardEncrypt/doc/sogood.html>

[Internet04] “Integer factorization”, [http://en.wikipedia.org/wiki/Integer\\_factorization](http://en.wikipedia.org/wiki/Integer_factorization)

[Internet05] “Thirty Years of Integer Factorization”, <http://algo.inria.fr/seminars/sem00-01/morain.html>

## 6. ÍNDICE

<b>SUMÁRIO</b> .....	<b>1</b>	3.1 A COMPLEXIDADE .....	4
<b>PALAVRAS-CHAVE</b> .....	<b>1</b>	3.2 OS ALGORITMOS .....	5
<b>1. INTRODUÇÃO</b> .....	<b>1</b>	3.2.1 Algoritmo Pesquisa Directa .....	5
<b>2. FACTORIZAÇÃO DE NÚMEROS INTEIROS</b> .....	<b>1</b>	3.2.2 Algoritmo de Fermat .....	5
2.1 OS NÚMEROS PRIMOS .....	1	3.2.3 Algoritmo da Diferença dos Quadrados .....	5
2.2 QUANTOS PRIMOS EXISTEM? .....	2	3.2.4 Algoritmo de Euler .....	5
2.3 O CRIVO DE ERATOSTHENES .....	2	3.2.5 Algoritmo Pollard rho - $\rho$ .....	6
2.4 OUTROS MÉTODOS PARA OBTEN NÚMEROS PRIMOS 2		3.2.6 Algoritmo Pollard p-1 .....	6
2.4.1 Um teorema chinês .....	2	3.2.7 Algoritmo Williams p+1 .....	6
2.4.2 Primos de Mersenne .....	2	3.2.8 Algoritmo das Curvas Elípticas .....	6
2.5 A IMPORTÂNCIA DOS NÚMEROS PRIMOS .....	3	3.2.9 Algoritmo da Curva Elíptica de Lenstra .....	6
2.6 FACTORIZAÇÃO DE NÚMEROS .....	3	3.2.10 Algoritmo das Frações Continuadas .....	6
2.6.1 Factos históricos .....	3	3.2.11 Algoritmo do Crivo Quadrático .....	7
2.7 QUAL CLASSE DE COMPLEXIDADE DA FACTORIZAÇÃO DE NÚMEROS INTEIROS .....	4	3.2.12 Algoritmo de Legendre .....	7
<b>3. ALGORITMOS PARA FACTORIZAR NÚMEROS INTEIROS</b> .....	<b>4</b>	3.2.13 Algoritmo do Crivo de Campo Numérico .....	7
		3.3 RESUMO DAS COMPLEXIDADES .....	7
		<b>4. CONCLUSÕES</b> .....	<b>7</b>
		<b>5. REFERÊNCIAS</b> .....	<b>8</b>
		<b>6. ÍNDICE</b> .....	<b>9</b>