Método de Fermat – Exemplo

Por exemplo: seja $n = 2027651281$; $a = \lfloor \sqrt{n} \rfloor + 1 = 45030$ é o primeiro valor a testar.

	a	b		a	b
1º	45030	$\sqrt{45030^2 - 2027651281} = 222{,}75$	**7º**	45036	$\sqrt{45036^2 - 2027651281} = 768{,}12$
2º	45031	$\sqrt{45031^2 - 2027651281} = 373{,}73$	**8º**	45037	$\sqrt{45037^2 - 2027651281} = 824{,}67$
3º	45032	$\sqrt{45032^2 - 2027651281} = 479{,}31$	**9º**	45038	$\sqrt{45038^2 - 2027651281} = 877{,}58$
4º	45033	$\sqrt{45033^2 - 2027651281} = 565{,}51$	**10º**	45049	$\sqrt{45039^2 - 2027651281} = 927{,}49$
5º	45034	$\sqrt{45034^2 - 2027651281} = 640{,}21$	**11º**	45040	$\sqrt{45040^2 - 2027651281} = 974{,}84$
6º	45035	$\sqrt{45035^2 - 2027651281} = 707{,}06$	**12º**	45041	$\sqrt{45041^2 - 2027651281} = 1020$

Concluindo:

$$n = 45041^2 - 1020^2, \quad n = 46061 \times 44021$$

Estudo Comparativo dos Vários Métodos

Fazendo um estudo comparativo dos vários métodos temos:

n	Factores		Divisões	Euclides	Fermat
1457	47	31	0,002s	0,015s	0,001s
13199	197	67	0,005s	0,030s	0,002s
281161	3559	79	0,006s	0,092s	0,218s
701123	3559	197	0,016s	0,169s	0,179s
23420707	41017	571	0,047s	0,839s	2,728s
488754769	110503	4423	0,361s	4,477s	6,093s
2027651281	46061	41017	3,611s	19,575s	0,004s
103955963689	47188363	2203	0,179s	51,891s	3926,6s
128228613281	58206361	2203	0,180s	58,180s	17175,0s
210528952589	95564663	2203	0,182s	75,888s	7953,8s
2746662891777043	47188363	58206361	3861,6s	—	50,333s
4509540007616669	47188363	95564663	3857,9s	—	712,73s

Todos os valores respeitam a testes efectuados sob as mesmas condições computacionais: sistema GNU/Linux 2.6.8; Intel Pentium 4 a 3,0GHz; 2GiB RAM; Octave 2.1.69. Os tempos referem-se ao tempo gasto pelo processador tal como nos é dado pelo sistema operativo.

Criptoanálise RSA

Para evitar a utilização de algoritmos de factorização conhecidos, no método criptográfico RSA, p e q devem ser escolhidos satisfazendo algumas condições.

Algoritmo	Restrição sobre p e q
Divisões (Crivo de Eratóstenes)	p e q devem ser «grandes»
Fermat	$\lvert p - q \rvert$ deve ser «grande»
Pollard $p-1$	$p-1$ e $q-1$ devem ser múltiplos de, pelo menos, uma potência «grande» de um número primo
Curvas elípticas	p e q devem ter aproximadamente o mesmo número de bits
Crivo quadrático	p e q devem ter, pelo menos, 1024 bits cada

ElGamal — Geração de Chaves

A segurança da cifra ElGamal está baseada na intractibilidade do problema do logaritmo discreto.
A geração de chaves segue o padrão de geração de chaves das cifras de chave pública, cada entidade gera um par de chaves, e concordam numa forma de distribuir as chaves públicas.

Geração de Chaves

Cada entidade A deve proceder do seguinte modo:

1. Gera de forma aleatório um primo de grande dimensão, e g, uma raiz primitiva módulo p (algoritmo H4.84).

Definição (Raiz Primitiva)

g é uma raiz primitiva módulo n se todo número a coprimo com n for congruente com uma potência de g módulo n.
Ou seja, g é uma raiz primitiva módulo n se para cada inteiro a coprimo com n, existe um inteiro k tal que $g^k \equiv a \pmod{n}$. Esse valor k é chamado de índice ou logaritmo discreto de a para a base g módulo n.

ElGamal — Geração de Chaves

1. Seleccionar aleatoriamente um inteiro a, $1 \leq a \leq p-2$ e calcular $r \equiv g^a \pmod{p}$ (algoritmo H2.143).
2. A chave pública de A é (p, g, r);
   A chave privada de A é a.

ElGamal — Encriptação

Na encriptação ElGamal (como em todas as cifras deste tipo) uma entidade B encripta a mensagem para outra entidade A, com a chave pública de A. Depois A desencripta a mensagem através da sua chave privada.

Algoritmo ElGamal — Encriptação

B deve proceder do seguinte modo:

1. obtêm a chave pública de A, (p, g, r), com $r = g^a$.
2. Representa a mensagem como um inteiro m na gama $\{0, 1, \ldots, p-1\}$.
3. Selecciona aleatoriamente um inteiro k, $1 \leq k \leq p-2$.
4. Calcular $\gamma \equiv g^k \pmod{p}$ e $\delta \equiv m \cdot r^k \pmod{p}$.
5. O texto cifrado é $c = (\gamma, \delta)$.

ElGamal — Desencriptação

Algoritmo ElGamal — Desencriptação

Para recuperar o texto claro m, A deve proceder do seguinte modo:

1. Utilizando a sua chave privada a calcular $\gamma^{p-1-a} \pmod{p}$ (nota: $\gamma^{p-1-a} = \gamma^{-a} = g^{-ak}$, em $\mathbb{Z}_p^*$).
2. Recupera m calculando $\gamma^{-a} \cdot \delta \pmod{p}$.

Pode-se verificar que a cifra ElGamal é uma cifra

$$\gamma^{-a} \cdot \delta \equiv g^{-ak} m g^{ak} \equiv m \pmod{p}$$

ElGamal — Um exemplo

Exemplo de Encriptação

A entidade A selecciona o primo $p = 2357$ e uma raiz primitiva módulo 2357, $g = 2$. Para chave privada A escolhe $a = 1751$ $(1 \leq a \leq p-2)$ e calcula

$$g^a \mod p = 2^{1751} \mod 2357 = 1185$$

A chave pública de A é $(p = 2357, g = 2, r = g^a = 1185)$
Para encriptar a mensagem $m = 2035$, B selecciona aleatoriamente um inteiro $k = 1520$ $(1 \leq k \leq 2355)$ e calcula

$$\gamma = 2^{1520} \mod 2357 = 1430$$

e

$$\delta = 2035 \cdot 1185^{1520} \mod 2357 = 697$$

A mensagem encriptada é então $c = (1430, 697)$

ElGamal — Um exemplo

Exemplo de Encriptação (Continuação)

Para desencriptar a mensagem A calcula

$$\gamma^{p-1-a} = 1430^{605} \mod 2357 = 872$$

A mensagem original m, é obtida calculando:

$$m = 872 \cdot 697 \mod 2357 = 2035$$

Parâmetros de Encriptação Comuns

Nota (Parâmetros de Encriptação Comuns)

Todas as entidades numa dada rede de comunicação podem decidir usar em comum o mesmo número primo p e a mesma raiz primitiva g, nesses casos tanto p como g deixam de fazer parte da chave pública.

Pode-se ter então chaves públicas de menor dimensão. Além disso as exponenciações que é necessário calcular podem ser alvo de uma pré-computação (H14.6.3), o que visto do ponto de vista da segurança significa que se torna necessário escolher valores para p maiores (≥ 768, de preferência 1024).

Eficiência da Encriptação

- O processo de encriptação requer duas exponenciações, módulo p, $g^k \mod p$ e $(g^a)^k \mod p$. O cálculo destas exponenciações podem ser melhorado seleccionando os expoentes k com um determinado tipo de estrutura.
- Uma desvantagem da cifra ElGamal é que tem um factor de expansão da mensagem cifrada de 2, isto é a o comprimento da mensagem cifrada é o dobro da mensagem clara.

Encriptação com Aleatoriedade

A cifra ElGamal é uma das cifra que usa aleatoriedade no seu algoritmo de encriptação. A ideia fundamental por detrás da utilização de aleatoriedade numa cifra é a de aumentar a segurança da cifra através da utilização de um, ou mais, dos métodos seguintes:

- Aumentar a dimensão dos espaço das mensagens claras;
- evitar, ou pelo menos diminuir, a eficácia dos ataques de texto claro escolhido (evitando uma correspondência de um para um entre os textos claros e os textos cifrado).
- evitar, ou pelo menos diminuir, a eficiência dos ataques através da análise estatística.