Criptografia Assimétrica

Introdução à Aritmética por trás dos algoritmos para Criptografia Assimétrica

Enno Nagel

Palestra no ICMC em São Carlos no dia 21 de Fevereiro de 2019

Criptografia Histórica

A criptografia estuda a transformação de um

texto inteligível

$$\Downarrow$$

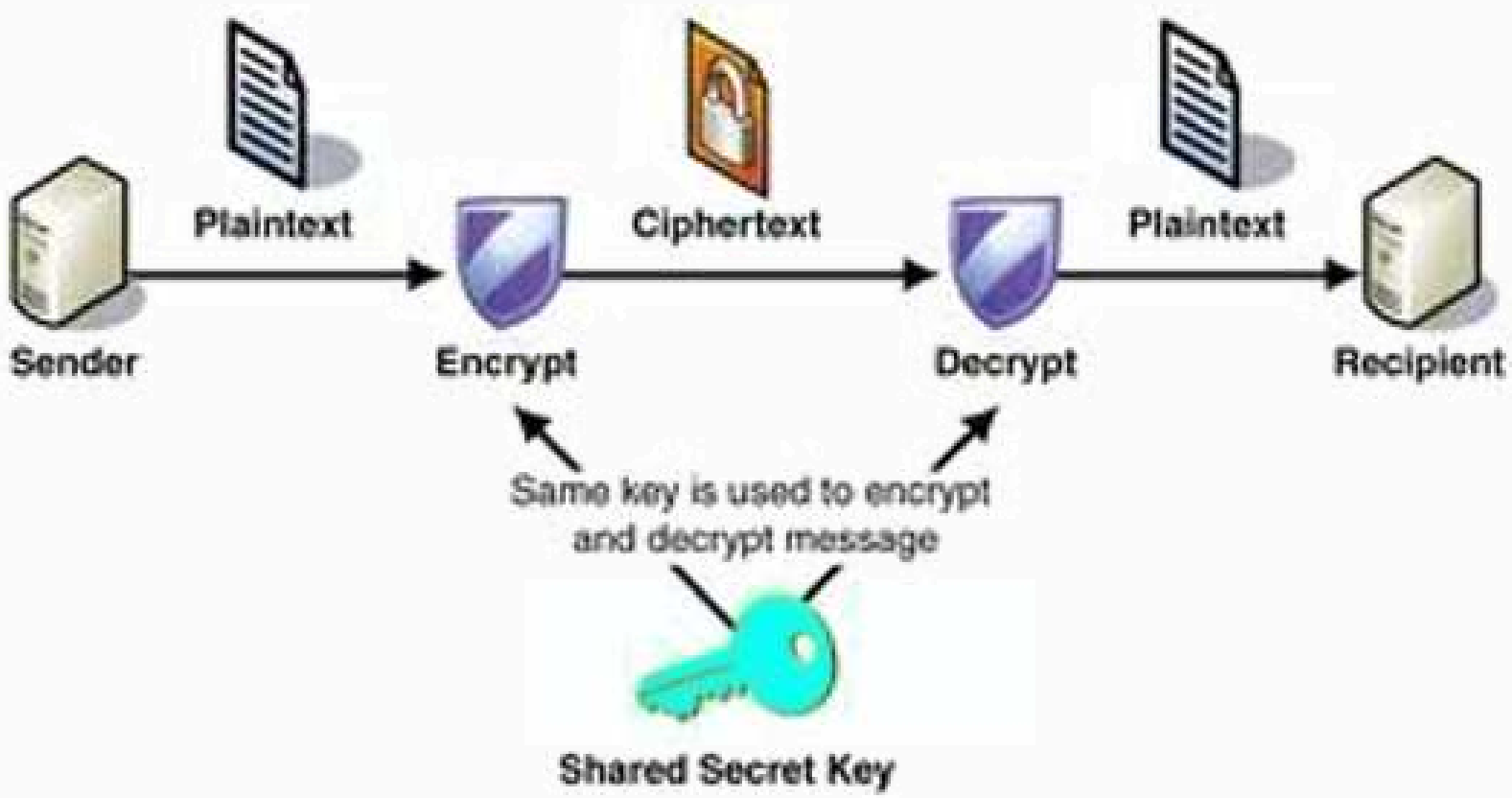
texto **in**inteligível

tal que só uma informação adicional secreta, a **chave**, permite desfazê-la.

Criptografia Simétrica

A criptografia é **simétrica** se

chave para cifrar = chave para decifrar.



A criptografia simétrica já era usada pelos egípcios 2000 anos antes de Cristo, e todos os exemplos históricos são simétricos.

Protótipos de Criptografia Simétrica Histórica

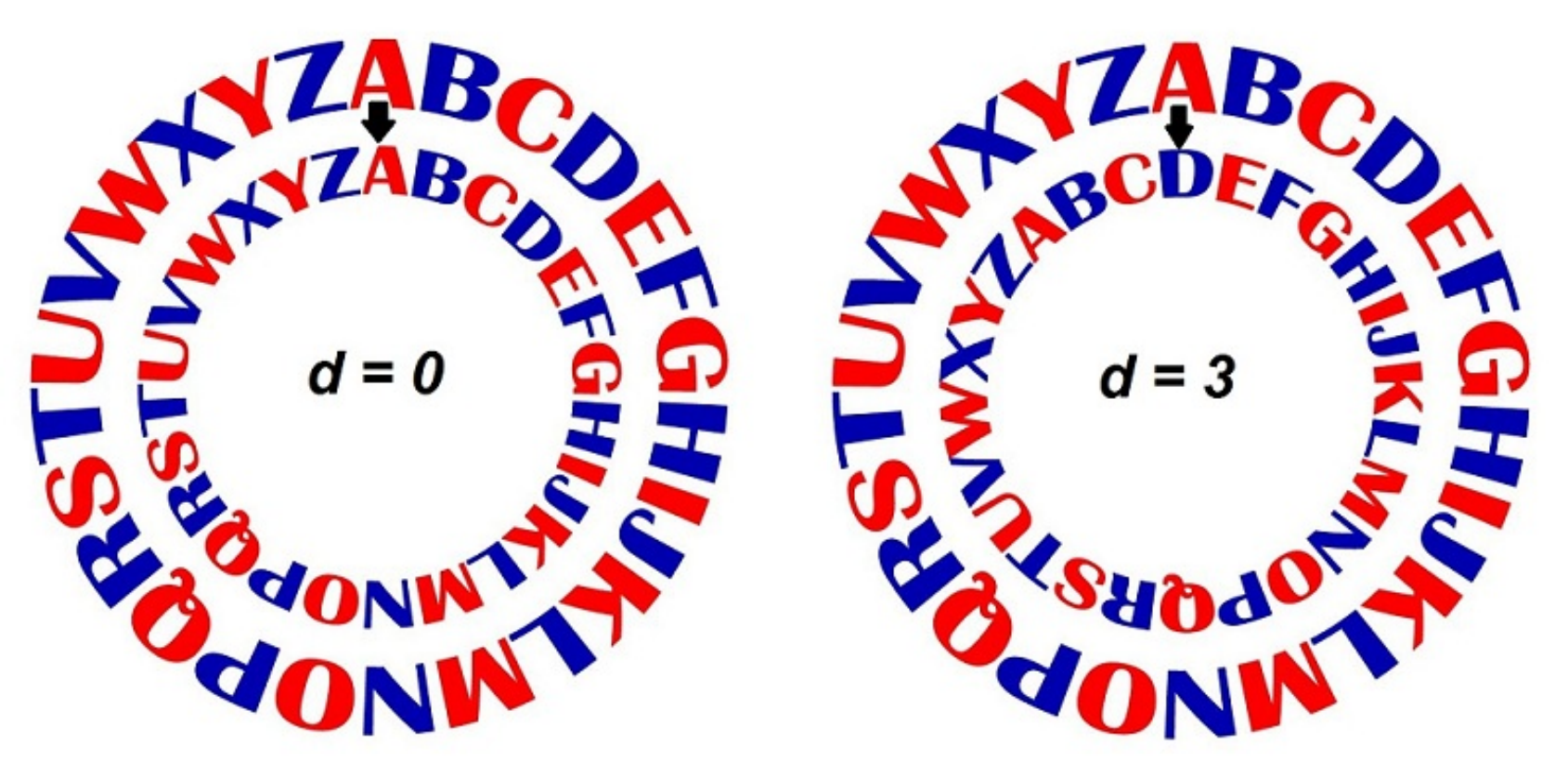


Figura 1: Se imaginemos que as letras do alfabeto formem duas rodas, então o algoritmo de César desloca uma delas por uma distância d fixa

Substituição do Alfabeto

Na **substituição**, as letras do **alfabeto** são permutadas:

- No César por traslado,

$$\mathrm{A} \mapsto \mathrm{D}, \mathrm{B} \mapsto \mathrm{E}, \mathrm{C} \mapsto \mathrm{F}, \ldots, \mathrm{W} \mapsto \mathrm{Z}, \mathrm{X} \mapsto \mathrm{A}, \ldots, \mathrm{Z} \mapsto \mathrm{C}.$$

- Em geral, por exemplo, nos cilindros da `Engima`, por permutação arbitrária,

A	B	...	Y	Z
↓	↓	...	↓	↓
E	Z	...	G	A

Figura 2: A cítala enrolada por um cinto de couro

Permutação do Texto (Claro)

Na **Permutação** (ou melhor transposição), as letras do **do Texto** (claro) são permutadas. Por exemplo, na cítala as duas linhas

l	u	a
m	e	l

tornam-se as três linhas

l	m
u	e
a	l

que são concatenadas à linha

L	M	U	E	A	L

Criptografia Moderna

A criptografia estuda a transformação de

dados inteligíveis

$$\Downarrow$$

dados **in**inteligíveis

tal que só uma informação adicional secreta, a **chave**, permite desfazê-la.

Outrora (na época análoga):

dados = textos.

Hoje (na época digital):

dados = arquivo digital (de texto, imagem, som, vídeo, ...)
= sequência de bits (= 0, 1)
= sequência de bytes (= 00, 01, ..., FE, FF)
= número (= 0, 1, 2, 3 ...).

Criptografia simétrica versus assimétrica

O ponto de vista dos dados

- como **sequência de bytes** é usado na criptografia **simétrica** moderna pelas redes de **substituição** e **permutação**, e
- o como **número** é usado na criptografia **assimétrica** que se baseia na **aritmética modular**.

Codificação de Textos

Texto inglês (= sequência de letras latinos)

A codificação **ASCII** cobre todas as letras inglesas (além dos símbolos de pontuação, por exemplo) e envia um símbolo a um byte. Por exemplo,

$$\mathrm{A} \mapsto 65, ..., \mathrm{Z} \mapsto 90; a \mapsto 97, ..., z \mapsto 122.$$

Texto Internacional (= sequência de símbolos)

A codificação **UTF-8** inclui a ASCII e cobre todas as letras de todos os idiomas (por exemplo, chinês, coreano, ... e além disso por exemplo todos os símbolos matemáticos).
Ela envia um símbolo a $1, 2, 3$ ou até 4 bytes, onde

$$\#\{\text{bytes}\} = \#\{\text{dígitos bin. consecutivos iniciais iguais a } 1\} + 1$$

Codificação de Imagens (por um *bitmap* = mapa de graus das cores primárias)

Uma imagem

- é um conjunto de pixeis (por exemplo, 1024×768),
- cada pixel é uma cor, e
- cada cor é os seus graus de intensidade da luz das três cores primárias
  - Vermelho (= *R*ed),
  - Verde (= *G*reen) e
  - Azul (= *B*lue).

  O homem não consegue distinguir mais de 256 graus de cada cor primária,

  $\Longrightarrow$ basta codificar cada pixel por três bytes.

Mapa de graus das cores primárias (RGB = Vermelho, Verde e Azul)

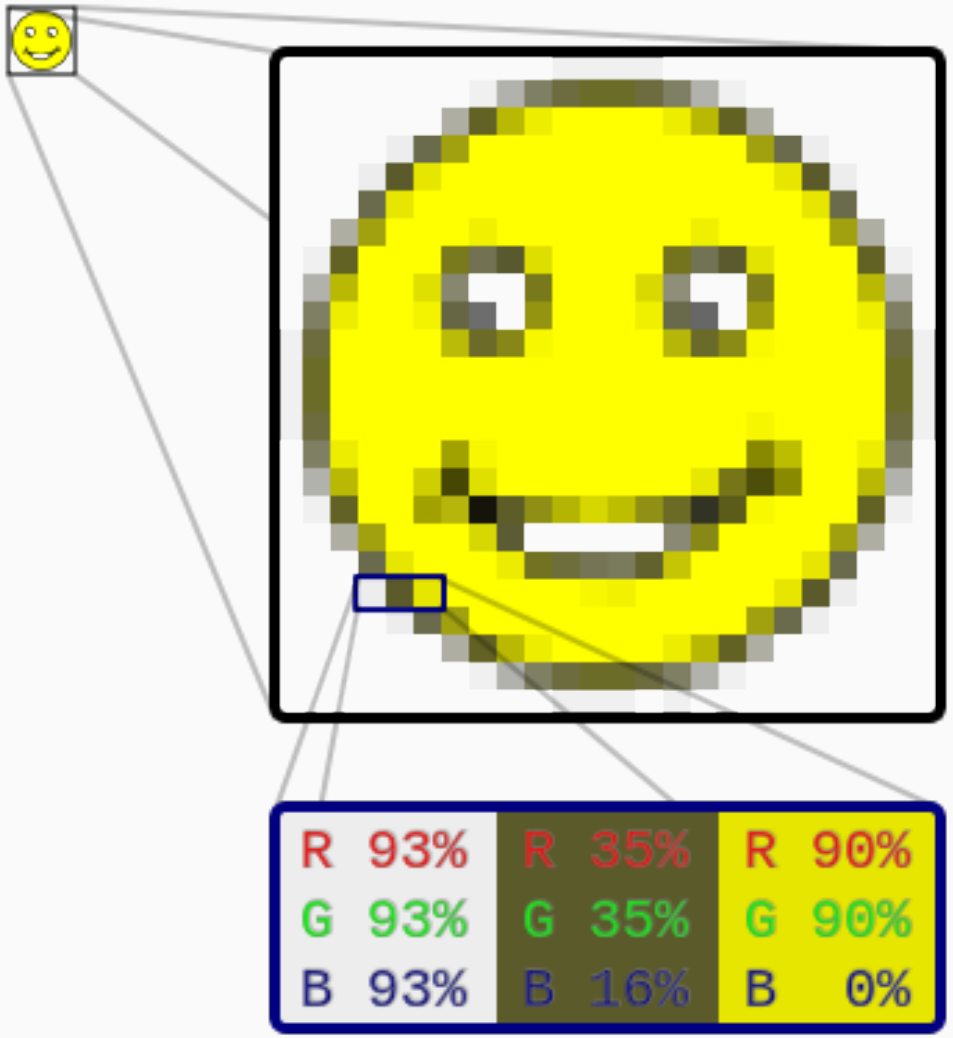
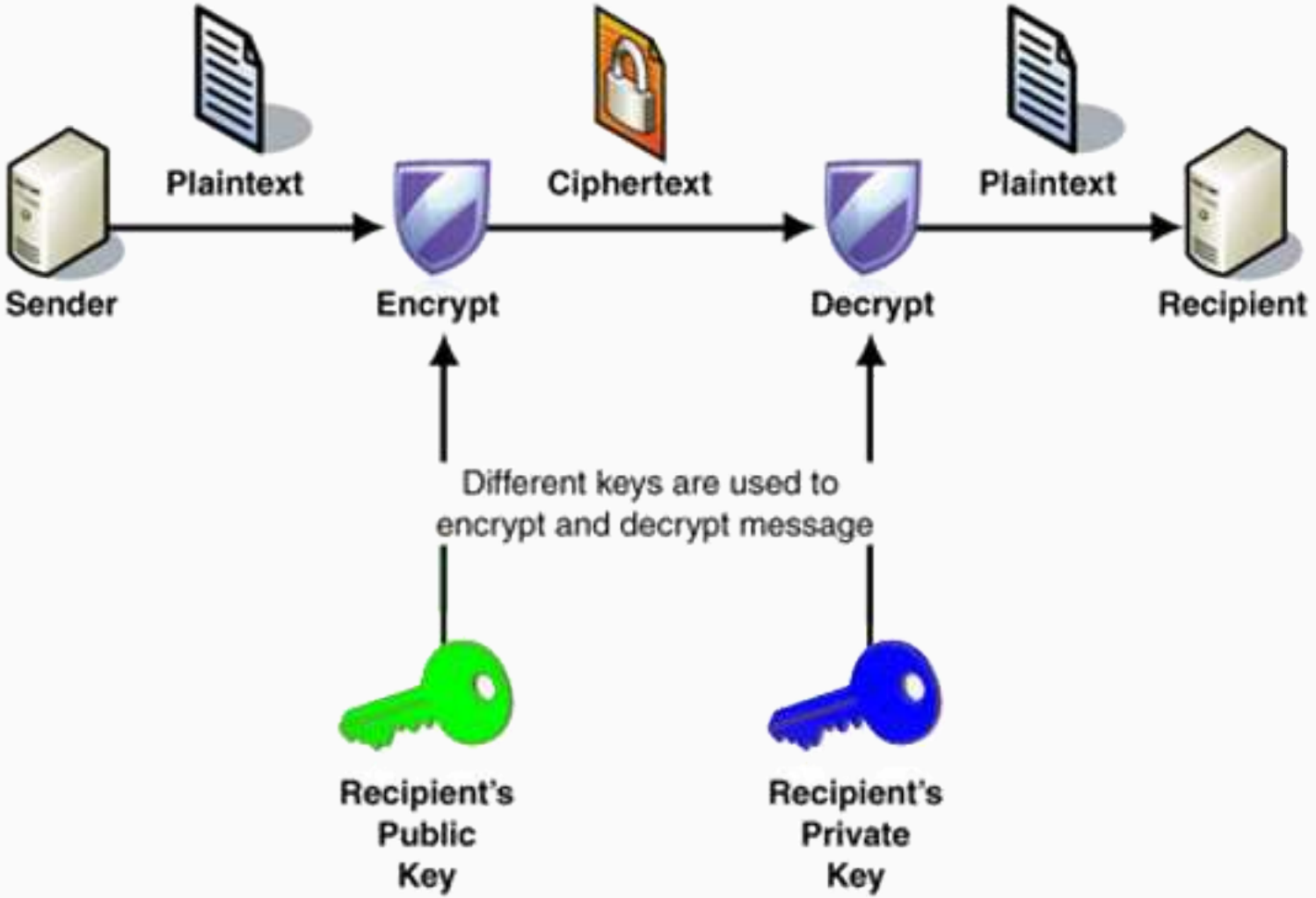


Figura 3: bitmap

Criptografia Assimétrica

A criptografia é **assimétrica** se

chave para cifrar ≠ chave para decifrar.



Chave pública e privada

Há duas chaves, uma chave **pública** e outra **privada**. Dois usos:

- (**Cifração**) A chave **pública** usa-se para **cifrar**, e a chave *privada* para *decifrar*,

  $\Longrightarrow$ pode ser **transferido** um texto por qualquer cifrador ao decifrador e *a ele só*.

- (**Assinatura Digital**) A chave **privada** usa-se para **cifrar**, e a chave *pública* para *decifrar*,

  $\Longrightarrow$ pode ser **verificado** um texto por qualquer decifrador se origina do cifrador e *dele só*.

Inventores da Ideia da Criptografia Assimétrica

A criptografia assimétrica foi sugerida pela primeira vez, publicamente, em Diffie e Hellman (1976). Antes (uns 20 anos), somente os serviços secretos tiveram consciência desta técnica.

Figura 4: Diffie e Hellman

RSA = Rivest, Shamir e Adleman

Com efeito, Diffie e Hellman (1976) introduziu apenas um protocolo para a criptografia assimétrica, mas não o pôs em prática. Isto foi feito pela primeira vez em Rivest, Shamir, e Adleman (1978), em que o algoritmo criptográfico RSA foi criado.

Figura 5: Os inventores do algoritmo RSA, Shamir, Rivest e Adleman

O Anel dos Números Inteiros

Denotem

$$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\} \quad \text{e} \quad \mathbb{R}$$

os *anéis* dos números inteiros e dos números reais (= a reta).

Anel

Um **anel** (comutativo com 1) é

- um conjunto que contém
  - $\mathbf{0}$ (= o elemento neutro da adição), e
  - $\mathbf{1}$ (= o elemento neutro da multiplicação);
- sobre o qual operam
  - a **adição** $+$ (e o seu *inverso* $-$), e
  - a **multiplicação** $\cdot$,

  que satisfazem a lei *associativa*, *comutativa* e *distributiva*.

Funções sobre Conjuntos Discretos

Na criptografia assimétrica,

- a *facilidade* de cifrar (um número), e
- a *dificuldade* de decifrar (um número)

baseiam-se em uma função invertível tal que

- ela é **facilmente** computável, mas
- a sua função inversa é **dificilmente** computável.

Exemplos para tais funções são

- a *exponenciação* $x \mapsto g^x$ (no algoritmo `ElGamal`), e
- a *potenciação* $x \mapsto x^e$ (no algoritmo RSA)

mas **não** definidas sobre $\mathbb{Z}$, porque $\mathbb{Z}$ é incluso em $\mathbb{R}$ e ambas as funções, exponenciação e potenciação, são **contínuas** sobre $\mathbb{R}$.

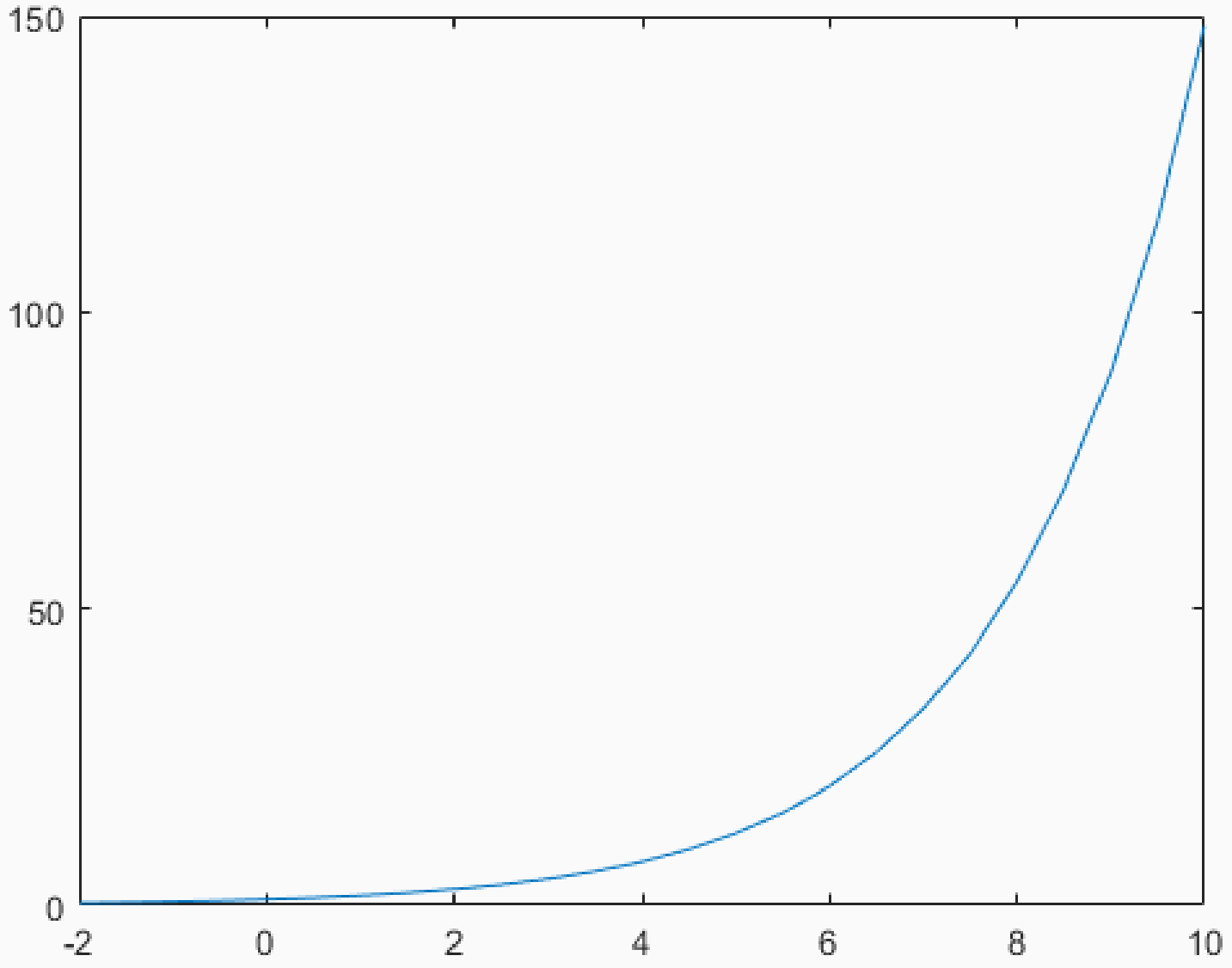


Figura 6: Exponencial $x \mapsto e^x$

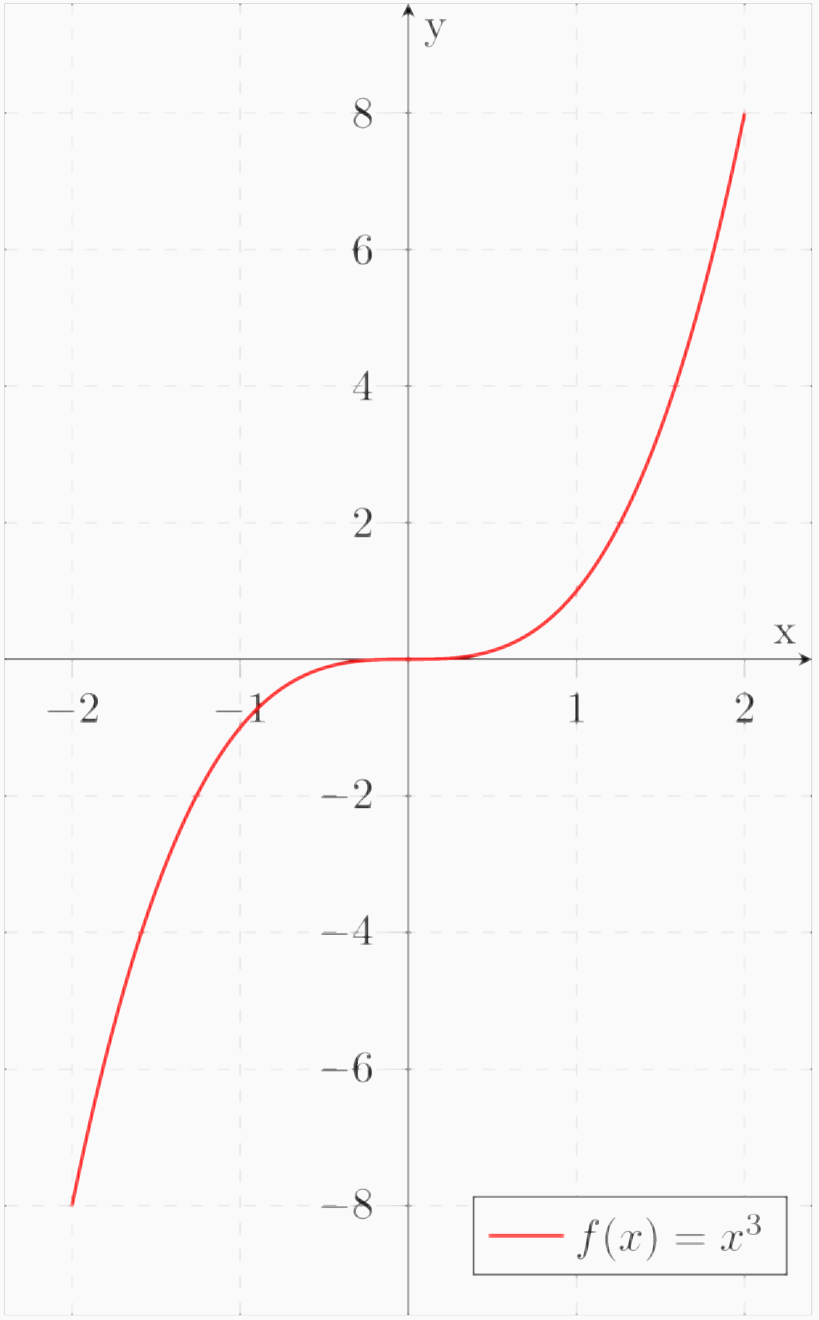


Figura 7: A parábola da função cúbica $x \mapsto x^3$

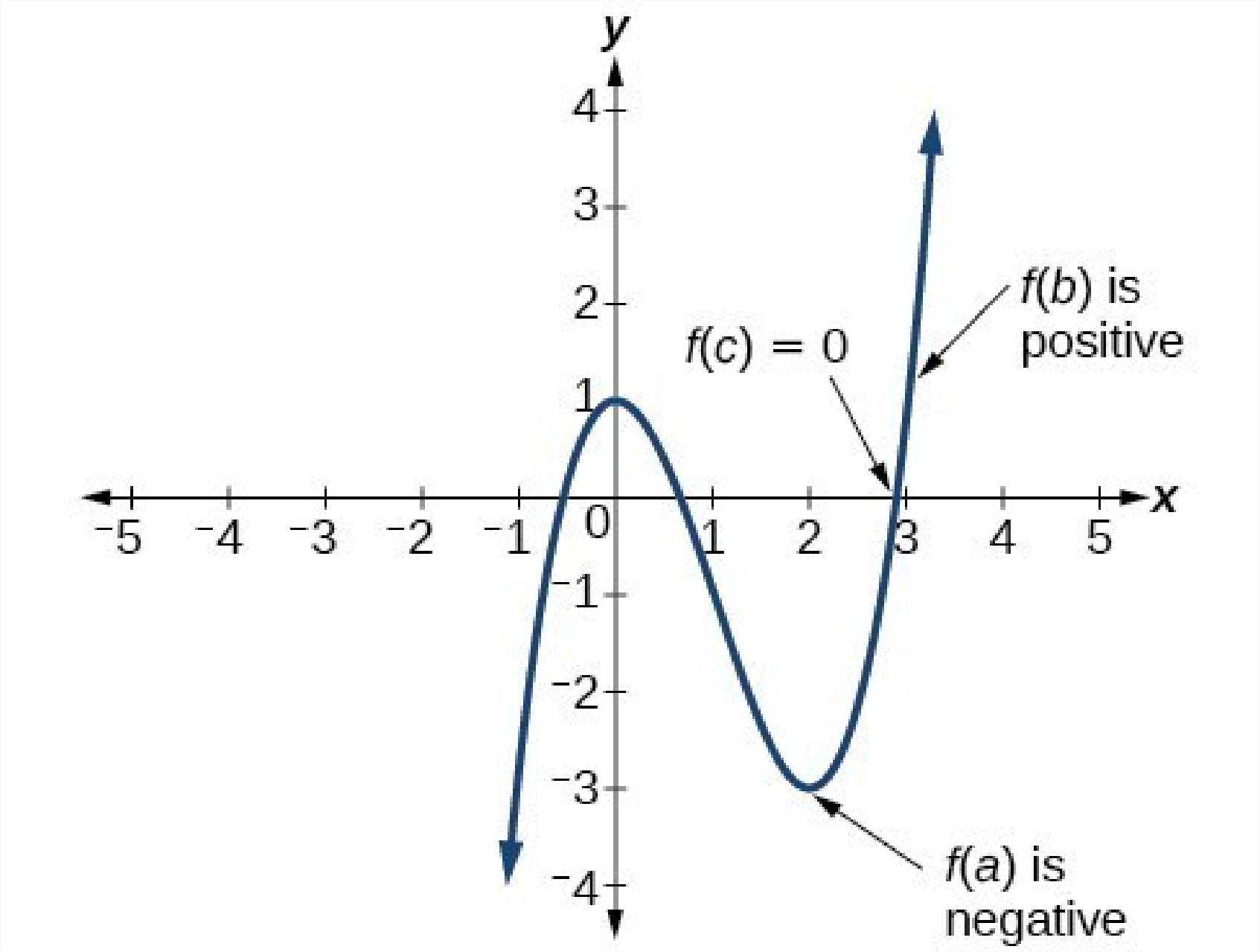


Figura 8: Teorema do Valor Intermédiario

Bissecção para encontrar o zero de uma função

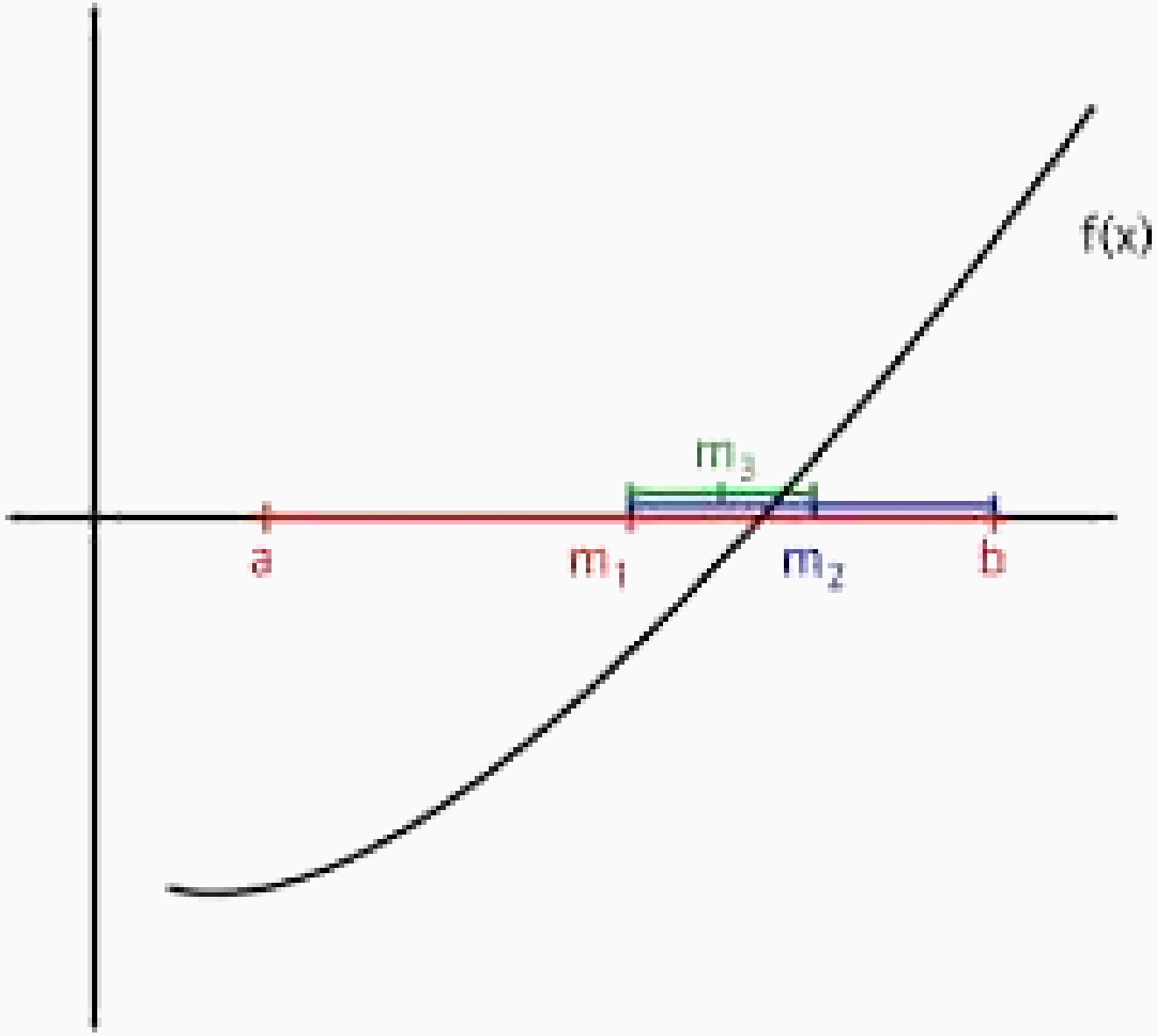


Figura 9: Bissecção de uma função contínua

Se estas funções fossem definidas sobre o anel $\mathbb{Z}$ e o qual é incluso em $\mathbb{R}$, os seus **inversos** podiam ser **aproximados** sobre $\mathbb{R}$, por exemplo, pelo **Método da Bisseção**: Dado y_0, encontrar um x_0 tal que $f(x) = y_0$ equivale encontrar um **zero** x_0 da função

$$x \mapsto f(x) - y_0.$$

1. (*Inicialização*) Escolha um intervalo $[x', x'']$ tal que

$$f(x') < 0 \quad \text{e} \quad f(x'') > 0.$$

2. (*Recalibração*) Calcule o seu meio $x''' := \frac{x'+x''}{2}$. Vale
   - ou $f(x''') = 0$, então $x''' = x_0$,
   - ou $f(x''') < 0$, então substituta a borda esquera x' por x''',
   - ou $f(x''') > 0$, então substituta a borda direita x'' por x'''.

   e itere com as novas bordas do intervalo.

Pelo **Teorema do Valor Intermediário**, o zero é garantido de estar no intervalo, que a cada passo diminui e converge à interseção.

Step	x	F(x)	|x(i) - x(i-1)|
x2	0	8	5
x3	-2.5	-18.875	2.5
x4	-1.25	-4.26563	1.25
x5	-0.625	1.42773	0.625
x6	-0.9375	-1.43726	0.3125
x7	-0.7813	-0.02078	0.15625
x8	-0.7031	0.69804	0.07813
x9	-0.7422	0.33745	0.03906
x10	-0.7617	0.15806	0.01953
x11	-0.7715	0.06857	0.00977
x12	-0.7764	0.02388	0.00488
x13	-0.7788	0.00154	0.00244
x14	-0.7800	-0.00962	0.00122

Figura 10: Bisseção do polinômio $\mathrm{F}(x) = x^3 + 3x^2 + 12x + 8$

Anéis Finitos

Para evitar a iterada aproximação de zeros e assim **dificultar** a computação do inverso (além de facilitar a computação da função), a criptografia assimétrica desenrola-se sobre os **anéis finitos**

$$\mathbb{Z}/m\mathbb{Z} = \{0, 1, ..., m-1\}.$$

Neles, vale $m = 1 + \cdots + 1 = 0$ e **toda adição** (e logo toda multiplicação e toda potenciação) **tem resultado** $< m$.

A adição de $\mathbb{Z}/m\mathbb{Z}$ é diferente da de $\mathbb{R}$; como anel **não** é simplesmente incluso! Por exemplo, para $m = 7$, vale

$$2^2 = 2 \cdot 2 = 4 \quad \text{e} \quad 3^2 = 3 \cdot 3 = 7 + 2 = 0 + 2 = 2.$$

Introduzamos estes anéis finitos primeiro pelo exemplo $\mathbb{Z}/12\mathbb{Z}$ (= o anel dos números das horas do relógio) e depois em geral.

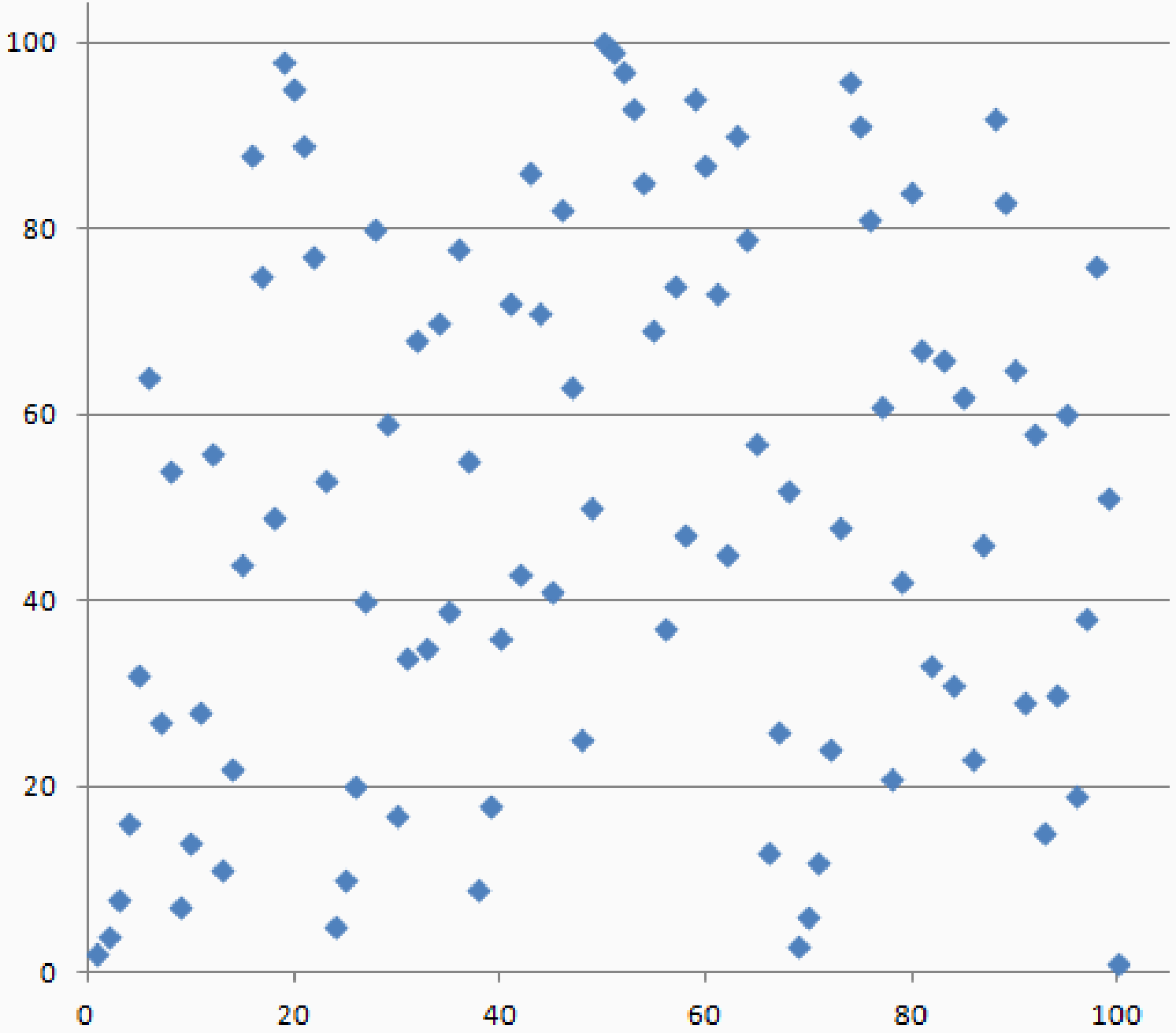


Figura 11: O gráfico de $y = 2^x$ sobre $\mathbb{Z}/101\mathbb{Z}$

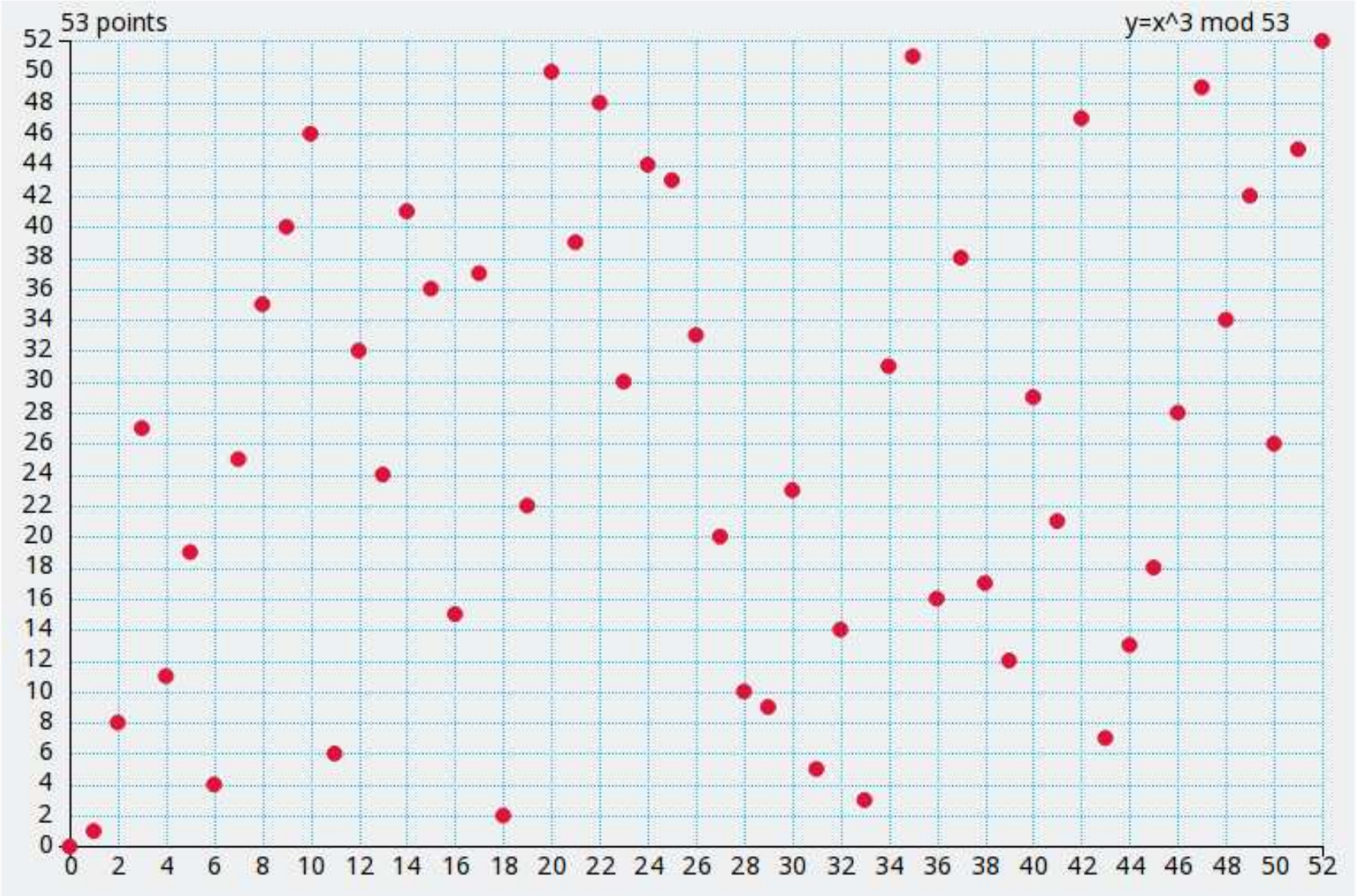


Figura 12: O gráfico de $y = x^3$ sobre $\mathbb{Z}/53\mathbb{Z}$

Relógio = Anel dos números $1, 2, ..., 11, 12 = 0$

Figura 13: O relógio como Anel dos números $1, 2, ..., 11, 12 = 0$

Aritmética do Relógio

O exemplo protótipo de aritmética modular é a aritmética do relógio, em que vale a equação

$$12 = 0,$$

e que implica, entre outros, as equações

$$9 + 4 = 1 \quad \text{e} \quad 1 - 2 = 11;$$

Quer dizer,

- 4 horas depois das 9 horas é 1 hora, e
- 2 horas antes da 1 hora são 11 horas.

Formalmente, derivamos estas equações das igualdades

$$9+4 = 13 = 12+1 = 0+1 = 1 \quad \text{e} \quad 1-2 = -1+0 = -1+12 = 11.$$

Podemos ir mais longe: $9 + 24 = 9$, quer dizer se agora são 9 horas, então 24 horas mais tarde também. Formalmente,

$$9 + 24 = 9 + 2 \cdot 12 = 9 + 2 \cdot 0 = 9.$$

Em geral, para quaisquer a e x em $\mathbb{Z}$,

$$a + 12 \cdot x = a$$

ou, equivalentemente, para quaisquer a e b em $\mathbb{Z}$,

$$a = b \quad \text{se } 12 \mid a - b$$

Congruência Modular

Seja $m \geq 1$ um inteiro. Os números inteiros a e b são **congruentes módulo** m ou, em fórmulas,

$$a \equiv b \mod m.$$

se $m \mid a - b$, isto é se a sua diferença $a - b$ é divisível por m. Ou, em outras palavras, se deixam o **mesmo resto** divididos por m.

O Anel $\mathbb{Z}/7\mathbb{Z}$ para $m = 7$

Figura 14: Os comprimidos semanais

O Anel $\mathbb{Z}/m\mathbb{Z}$ para $m = 15$

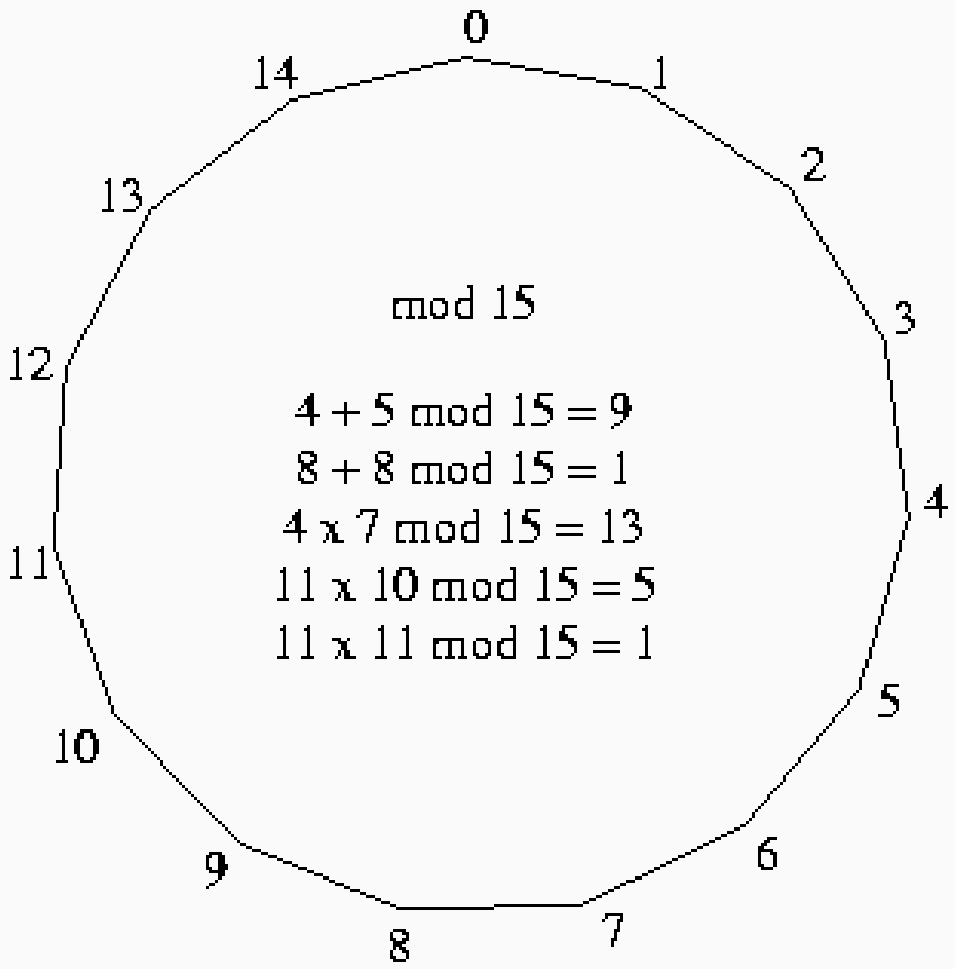


Figura 15: O relógio com 15 horas

O Anel Quociente

Dado um número inteiro m, construamos:

- o **menor** anel (= conjunto que contém 0 e 1 e sobre o qual $+$ e $\cdot$ operam), denotado por $\mathbb{Z}/m\mathbb{Z}$,
- com **uma aplicação** $\bar{\cdot}\colon \mathbb{Z} \to \mathbb{Z}/m\mathbb{Z}$ denotada

$$x \mapsto \bar{x}$$

que **satisfaz**

$$\overline{x+y} = \bar{x} + \bar{y}$$

(e portanto $\overline{x \cdot y} = \bar{x} \cdot \bar{y}$, em particular, $\bar{0} = 0$ e $\bar{1} = 1$), e

- tal que

$$x \mapsto 0 \iff m \mid x \qquad (\dagger)$$

ou, equivalentemente,

$$x \equiv y \bmod m \iff \bar{x} = \bar{y} \text{ em } \mathbb{Z}/m\mathbb{Z}.$$

Divisão Com Resto

Divisão Com Resto

Sejam a e b números inteiros positivos. Que a **dividido por** b tem **quociente** q e **resto** r significa

$$a = b \cdot q + r \quad \text{com } 0 \leq r < b.$$

Exemplo

Para $a = 230$ e $b = 17$, calculamos

$$230 = 17 \cdot 13 + 9.$$

Isto é, 230 *divido por* 17 tem *quociente* 13 e *resto* 9.

Construção

- como **conjunto**

$$\mathbb{Z}/m\mathbb{Z} := \{0, ..., m-1\};$$

e como **aplicação** $\bar{\cdot}\colon \mathbb{Z} \to \mathbb{Z}/m\mathbb{Z}$, o **resto** dividido por m,

$$x \mapsto r \quad \text{onde } x = qm + r \quad \text{com } r \text{ em } \{0, ..., m-1\};$$

- como **elementos neutros** da adição e multiplicação 0 e 1,
- como **operações** $+$ e $\cdot$ os **restos** da **soma** e do **produto** dividido por m,

$$x+y := r \quad \text{onde } x+y = qm+r \quad \text{com } r \text{ em } \{0, ..., m-1\}, \text{ e}$$

$$x \cdot y = r \quad \text{onde } x \cdot y = qm + r \quad \text{com } r \text{ em } \{0, ..., m-1\}.$$

Tabelas de Adição e Multiplicação para $m = 4$

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Potenciação Rápida (para `Diffie Hellman` e RSA)

Dados uma base b e um expoente e em $\mathbb{Z}$, para calcular

$$b^e \text{ em } \mathbb{Z}/\mathrm{M}\mathbb{Z},$$

1. Expande o expoente **binariamente**, isto é

   $$e = e_0 + e_1 2 + e_2 2^2 + \cdots + e_s 2^s \text{ com } e_0, e_1, ..., e_s \text{ em } \{0, 1\},$$

2. Calcula

   $$b^1, b^2, b^{2^2}, ..., b^{2^s} \bmod \mathrm{M}.$$

Como $b^{2^{n+1}} = b^{2^n \cdot 2} = (b^{2^n})^2$, isto é, cada potência é o **quadrado da anterior** (e **limitada** por M), cada uma, sucessivamente, é facilmente computável. Obtemos

$$b^e = b^{e_0 + e_1 2 + e_2 2^2 + \cdots + e_s 2^s} = b^{e_0} (b^2)^{e_1} (b^{2^2})^{e_2} \cdots (b^{2^s})^{e_s}$$

Isto é, somente as potências com expoentes $e_0, e_1, ..., e_s$ iguais a 1 importam, as outras podem ser omitidas.

Exemplo da Potenciação Rápida

Para calcular 3^{13} módulo 7, expandimos

$$13 = 1 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3$$

e calculamos, módulo 7,

$3^1 = 3, 3^2 = 9 \equiv 2, 3^{2^2} = (3^2)^2 \equiv 2^2 = 4,$ e $3^{2^3} = (3^{2^2})^2 \equiv 4^2 \equiv 2,$

obtendo

$$3^1 3 = 3^{1+2^2+2^3} = 3^1 \cdot 3^{2^2} \cdot 3^{2^3} = 3 \cdot 4 \cdot 2 \equiv 3 \mod 7.$$

Código Python

Calcula recursivamente x^y; para potências negativas usa $x^{-y} = 1/x^y$:

```python
# Function to calculate x raised to the power y
def power(x, y):

    if (y == 0): return 1
# round down y/2 and calculate x raised to y
    pow = power(x, int(y / 2))
    elif (int(y % 2) == 0):
        return pow * pow
    else:
        if(y > 0): return x * pow * pow
        else: return (pow * pow) / x
```

Para dificultar o cálculo dos inversos das funções criptográficas, precisaremos de grandes módulos primos. Existem:

Teorema de Euclides.

Existe uma **infinitude** de números primos. Isto é, há números primos **arbitrariamente grandes** (por exemplo, para o RSA serão precisos ≥ 2048 algarismos binários).

Demonstração (por Contradição):

Suponhamos o contrário, que haja só um número finito $p_1, ..., p_n$ de números primos. Considere

$$q = p_1 ... p_n + 1.$$

Como q é maior que $p_1, \ldots, p_n$, não é primo. Seja então p um número primo que divida q. Pela suposição, p em $\{p_1, ..., p_n\}$. Mas por definição q tem resto 1 divido por qualquer $p_1, \ldots, p_n$. Contradição! Logo, não existe um maior número primo. q.e.d.

Exemplos de Grandes Números Primos

Marin Mersenne (Oizé, 1588 — Paris, 1648) foi um padre mínimo francês que tentou encontrar, sem êxito, uma fórmula para todos os números primos. Motivada por uma carta de Fermat em que lhe sugeriu que todos os números $2^{2^p}+1$, os *Números de Fermat* fossem primos, Mersenne estudou os números

$$2^p-1 \quad \text{para } p \text{ primo .}$$

Em 1644 publicou o trabalho *Cogita physico-mathematica* que afirma que estes são primos para

$$p = 2, 3, 5, 7, 13, 17, 19, 31 \text{ e } 127.$$

(e erroneamente incluiu $p = 63$ e $p = 257$). Só um computador conseguiu mostrar em 1932 que $2^{257}-1$ é composto.

Recordes

Os números primos de Mersenne, da forma $2^p - 1$ para p primo, conhecidos são

2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281, 3217, 4253, 4423, 9689, 9941, 11213, 19937, 21701, 23209, 44497, 86243, 110503, 132049, 216091, 756839, 859433, 1257787, 1398269, 2976221, 3021377, 6972593, 13466917, 20996011, 24036583, 25964951, 30402457, 32582657, 37156667, 42643801, 43112609, 57885161, 74207281, 77232917 e 82589933

O número primo

$$2^{82\,589\,933} - 1$$

tem 24 862 048 algarismos. Foi encontrado no dia 8 de dezembro 2018 e é até hoje o **maior número primo conhecido**.

Detectar Números Primos

Um teste rápido se n é composto é o **Pequeno Teorema de Fermat**: Se há a tal que

$$a^n \not\equiv a \bmod n$$

então n é composto.

Mas o inverso não vale: Há números (que se chamam **Números de Carmichael**) que são compostos mas satisfazem, para todos os números a,

$$a^n \equiv a \bmod n.$$

O **teste de AKS** determina em tempo polinomial se n é composto ou primo (mais exatamente, em tempo $O(d)^6$ onde d = o número de dígitos d [binários] de n). Na prática, basta normalmente o **Teste de Miller-Rabin** probabilista que garante muito mais *testemunhas* (= números a que provam se n é composto ou não) do que o Pequeno Teorema de Fermat.

Teste por Números Primos

Os testes simplistas se um número n é primo ou não calculam os fatores de n e por isso são ineficientes.

Comparação com Fermat

Se para um número ímpar, um número possivelmente primo, escrevemos $n - 1 = 2^k q$, então pelo Teste de Fermat n não é primo se existe um inteiro a tal que $a^{2^k q} \not\equiv 1 \mod n$. O Teste de Miller-Rabin explicita $a^{q2^k} = (a^q)^{2^k} = ((a^q)^2) \cdots)^2 \not\equiv 1$.

O Teste de Miller-Rabin

Seja n ímpar e $n - 1 = 2^k q$ para números k e q (com q ímpar). Um inteiro a relativamente primo a a é uma **Testemunha de Miller-Rabin** (para a composição) de n, se

- $a^q \not\equiv \pm 1 \mod n$ e
- $a^{2q} a^{2^2 q}, ..., a^{2^{k-1} q} \not\equiv -1 \mod n$.

Quão provável que erramos?

Questão: Quais as chances que declaramos pelo Teste de Miller-Rabin acidentalmente um número primo, isto é, um número que é na verdade composto?

Teorema (sobre a frequência das testemunhas)

Seja n ímpar e composto. Então pelo menos 75% dos números em $\{1, ..., n-1\}$ são Testemunhas de Miller-Rabin para n.

Probabilidade

Logo, já depois de 5 tentativas $a_1, a_2, \ldots, a_5$ sem testemunha sabemos com uma chance $1/4^5 = 1/1024 < 0,1\%$, que o número é primo!

Inventores da Troca de Chaves

A criptografia assimétrica foi sugerida pela primeira vez, publicamente, em Diffie e Hellman (1976). Antes (uns 20 anos), somente os serviços secretos tiveram consciência desta técnica.

Figura 16: Diffie e Hellman

Troca de Chaves de Diffie-Hellman

Para Alice e Bob construírem uma chave secreta através de um canal inseguro, combinam em primeiro lugar

- um número primo p *apropriado*, e
- um número natural g *apropriado*.

e depois

1. Alice, para gerar **uma metade** da chave, escolhe a,
   - calcula $\mathrm{A} \equiv g^a \bmod p$, e
   - transmite A ao Bob.
2. Bob, para gerar **outra metade** da chave, escolhe b,
   - calcula $\mathrm{B} \equiv g^b \bmod p$, e
   - transmite B à Alice.
3. A chave secreta **mútua** entre Alice e Bob é

$$c := \mathrm{A}^b = (g^a)^b = g^{ab} = g^{ba} = (g^b)^a = \mathrm{B}^a \bmod p.$$

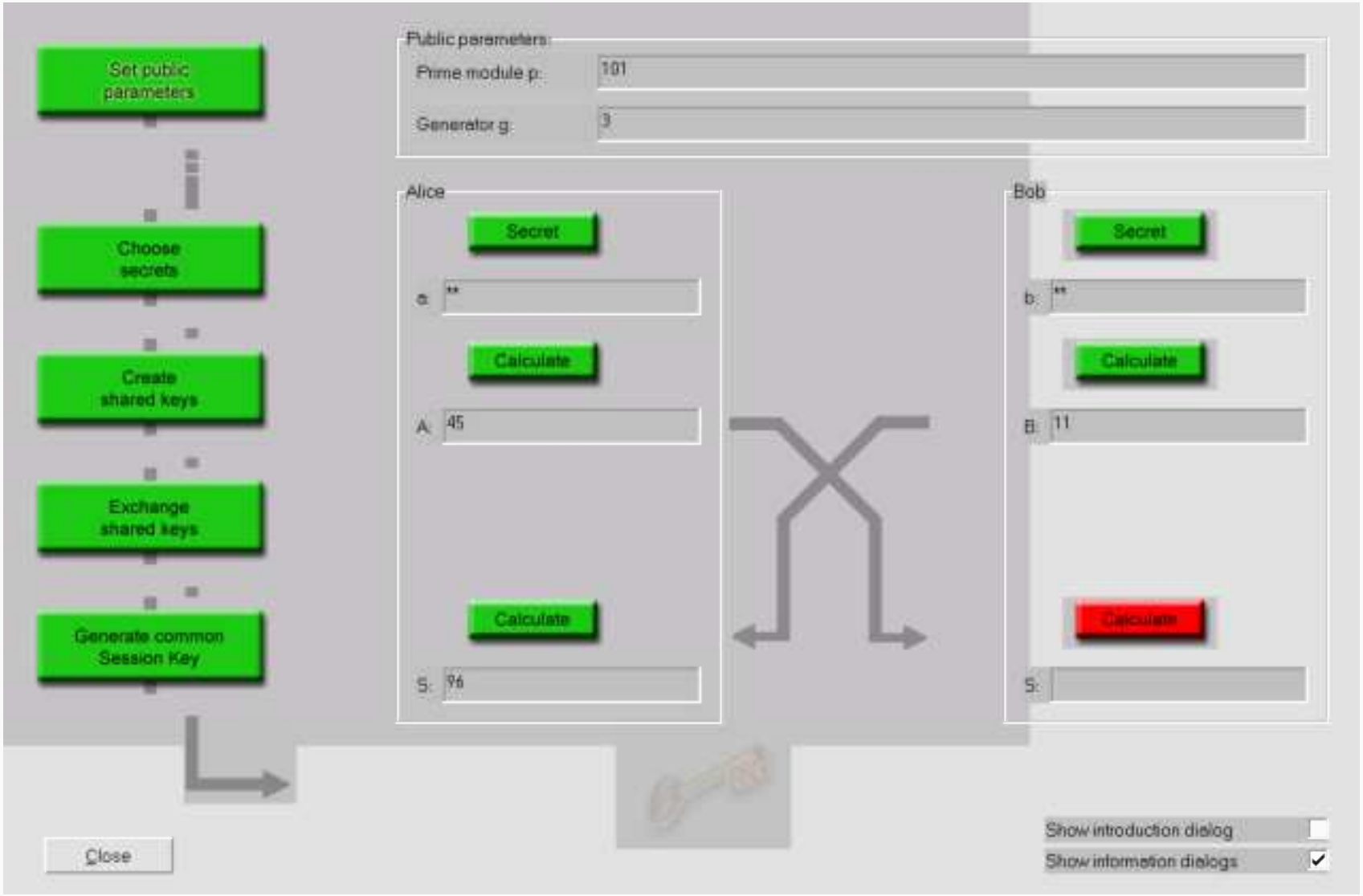


Figura 17: O `CrypTool 1` oferece no Menu `Individual Procedures -> Protocols` uma entrada para experimentar com os valores das chaves no `Diffie-Hellman`

Dificuldade = Computar o Logaritmo módulo p

Um olheiro obteria a chave secreta $c = \mathrm{A}^b = \mathrm{B}^a$ a partir de A e B, se pudesse computar

$$a = \log_g \mathrm{A} \quad \text{ou} \quad b = \log_g \mathrm{B} \mod p;$$

isto é, o *logaritmo* $\log_g$ inverte a *potenciação* $x \mapsto g^x = y$,

$$\log_g y = x \quad \text{com } x \text{ tal que } g^x = y.$$

Enquanto a potenciação é facilmente computável, o **logaritmo é dificilmente computável** para escolhas de p e g **apropriadas**:

- o número primo p
  - seja **grande** e
  - haja um número primo **grande** tem de dividir $p - 1$.
- as potências da base g gerem um **grande** conjunto (finito)

$$\{g, g^2, g^3, \ldots\}.$$

Números Apropriados

O Teorema de Euclides garante que haja números primos arbitrariamente **grandes** (> 1024 bits),

$$\#\{ \text{números primos} \} = \infty$$

e quase todo número primo p satisfaz que

- haja um número primo **grande** (> 768 bits) que divide $p - 1$.

O Teorema da *Raiz Primitiva* garante que sempre haja um número g em $\mathbf{F}_p^*$ tal que

$$\mathbf{F}_p^* = \{1, 2, 3, ..., p-1\} = \{g, g^2, g^3, ..., g^{p-1}\},$$

Em particular, se p é grande, então o conjunto gerado pelas potências da base g é **grande** (= $p - 1 \geq 1024$ bits).

Na prática, os números p e g são adotados duma fonte confiável.

O Criador do Algoritmo ElGamal

Figura 18: Taher El Gamal (*1955)

O Algoritmo ElGamal

Para a Alice enviar a mensagem m secretamente ao Bob através de um canal inseguro, combinam em primeiro lugar

- um número primo p *apropriado*, e
- um número natural g *apropriado*.

1. Bob, para **gerar** uma chave, escolhe um número b,
   - calcula $\mathrm{B} = g^b \bmod p$, e
   - transmite B à Alice.
2. Alice, para **cifrar**, escolhe um número a,
   - calcula $\mathrm{A} = g^a \bmod p$,
   - calcula $\mathrm{C} = \mathrm{B}^a \bmod p$ e $\mathrm{M} = m\mathrm{C} \bmod p$, e
   - transmite A e M à Alice.
3. Bob, para **decifrar**,
   - calcula $\mathrm{A}^b = (g^a)^b = g^{ba} = g^{ab} = (g^b)^a = \mathrm{B}^a = \mathrm{C} \bmod p$
   - calcula $\mathrm{C}^{-1} \bmod p$, e
   - calcula $\mathrm{M}\mathrm{C}^{-1} = m\mathrm{C}\mathrm{C}^{-1} = m \bmod p$.

Juiz e Matemático leigo francês Pierre de Fermat

Figura 19: Pierre de Fermat ($^{\dagger}1655$)

Pequeno Teorema de Fermat

Se p é primo, e p não divide a, então $a^{p-1} \equiv 1 \mod p$.

Demonstração:

Como $\#\{1, ..., p-1\} = p-1$ é finito, existe n e $m \leq p-1$ tal que $a^n = a^{n+m}$. O algoritmo de Euclides (a seguir) mostrará (porque p é primo) que podemos dividir por a. Logo $a^m = 1$. Pelo *Teorema de Lagrange* (omitido), m divide $p-1$!

Exemplo

Por exemplo, em $\mathbb{Z}/5\mathbb{Z}$ vale

$$1^4 = 1, 2^4 = 16 \equiv 1, 3^4 = 81 \equiv 1 \text{ e } 4^4 = (2^2)^4 = (2^4)^2 \equiv 1,$$

enquanto em $\mathbb{Z}/4\mathbb{Z}$ vale $2^3 = 8 \equiv 0$. Concluímos que 4 não é primo (sem conhecer seus fatores!).

O Matemático suíço Leonhard Euler

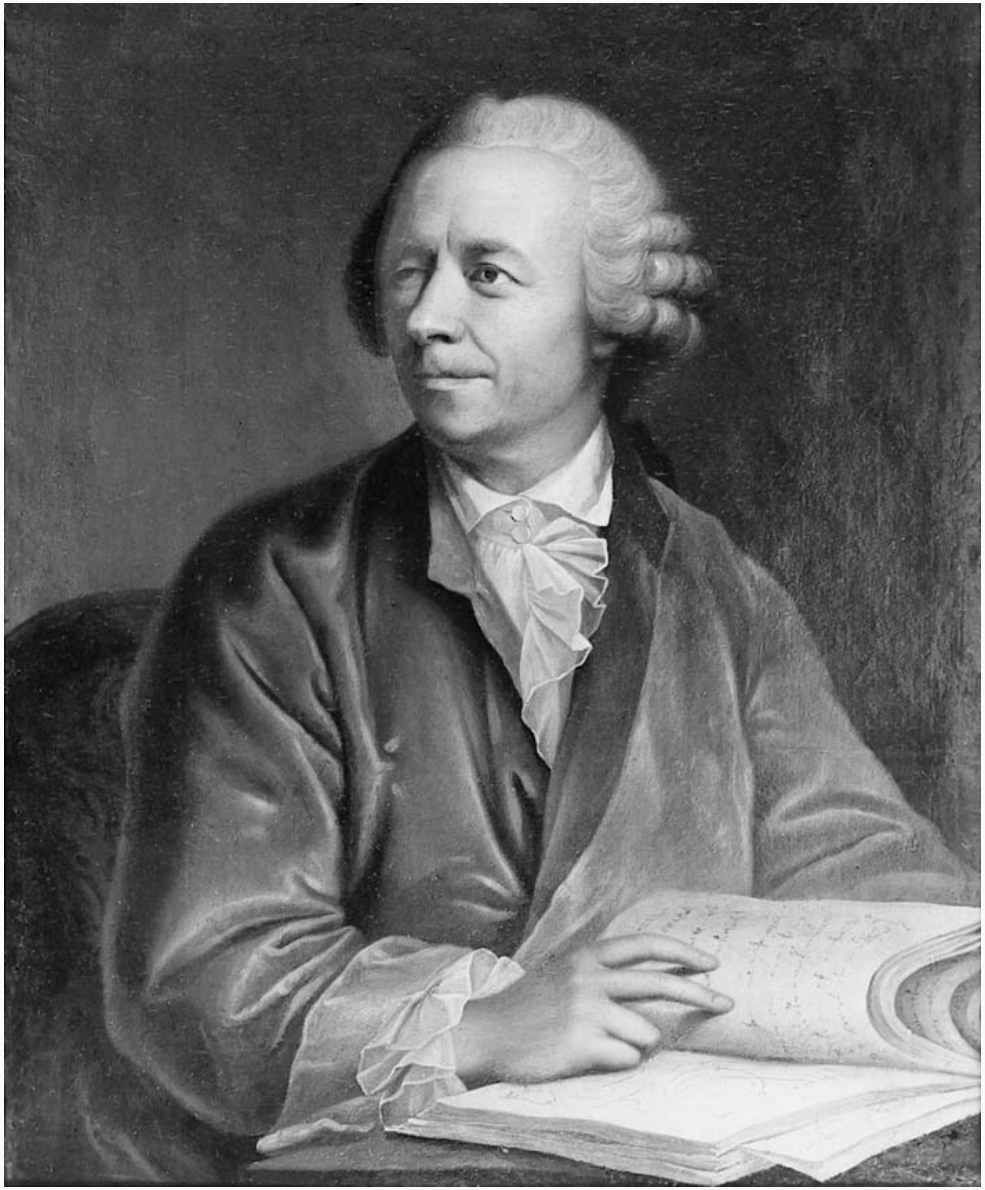


Figura 20: Leonhard Euler (1707 – 1783)

Fórmula de Euler

Sejam p e q números primos diferentes. Se a é divisível nem por p, nem por q, então

$$a^{(p-1)(q-1)} \equiv 1 \bmod pq.$$

Demonstração

Pelo pequeno teorema de Fermat,

$$a^{(p-1)(q-1)} = (a^{p-1})^{q-1} \equiv 1^{q-1} = 1 \bmod p$$

e

$$a^{(q-1)(p-1)} = (a^{q-1})^{p-1} \equiv 1^{p-1} = 1 \bmod q$$

isto é, p e q dividem $a^{(p-1)(q-1)} - 1$.

Sendo p e q primos diferentes, $\implies$ pq divide $a^{(p-1)(q-1)} - 1$, isto é, $a^{(p-1)(q-1)} \equiv 1 \bmod pq$.

Para números primos diferentes p e q, denotem

$$\mathrm{N} := pq \quad \text{e} \quad \phi(\mathrm{N}) := (p-1)(q-1).$$

Pela Fórmula de Euler $a^{\phi(\mathrm{N})} \equiv 1 \bmod \mathrm{N}$. Logo

- vale $a^{1+\phi(\mathrm{N})} = a^1 \cdot a^{\phi(\mathrm{N})} \equiv a^1 \cdot 1 = a \bmod \mathrm{N}$,
- vale $a^{1+2\phi(\mathrm{N})} = a^1 \cdot (a^{\phi(\mathrm{N})})^2 \equiv a^1 \cdot 1^2 = a \bmod \mathrm{N}$,
- vale $a^{1+3\phi(\mathrm{N})} = a^1 \cdot (a^{\phi(\mathrm{N})})^3 \equiv a^1 \cdot 1^3 = a \bmod \mathrm{N}, \ldots \implies$

Corolário

Sejam p e q números primos diferentes. Para qualquer expoente m tal que

$$m \equiv 1 \bmod \phi(\mathrm{N})$$

vale

$$a^m \equiv a \bmod \mathrm{N} \quad \text{para todo inteiro } a.$$

Observação **Crucial** para o algoritmo RSA

Se $m \equiv 1 \mod \phi(\mathrm{N})$, então pela *Fórmula de Euler* $a^m \equiv a \mod \mathrm{N}$, isto é, a potenciação é a identidade,

$$\cdot^m \equiv \mathrm{id} \mod \mathrm{N}$$

Em particular, se $m = \mathrm{E}d$ é o produto de dois números inteiros E e d, isto é,

$$\mathrm{E}d \equiv 1 \mod \phi(\mathrm{N}),$$

então

$$a = a^m = a^{\mathrm{E}d} = (a^{\mathrm{E}})^d.$$

Isto é, a potenciação $\cdot^d = \cdot^{1/\mathrm{E}}$ inverte $\cdot^{\mathrm{E}}$ módulo N, a **extração da E-ésima radiciação é a d-ésima potência**,

$$\sqrt[\mathrm{E}]{\cdot} = \cdot^{1/\mathrm{E}} \equiv \cdot^d \mod \mathrm{N}.$$

Calcular uma potência é muitíssimo mais fácil do que uma raiz!

Exemplo

Se $p = 3$ e $q = 11$, então

$$\mathrm{N} = pq = 33 \quad \text{e} \quad \phi(\mathrm{N}) = (p-1)(q-1) = 20.$$

Se $d = 3$ e $\mathrm{E} = 7$, então $m = \mathrm{E}d = 21 \equiv 1 \bmod 20$.

Por exemplo, para a base 2, verificamos

$$2^{\mathrm{E}} = 2^7 = 128 = 29 + 3 \cdot 33 \equiv 29 \bmod \mathrm{N}$$

e

$$29^d = 29^3 = (-4)^3 \equiv -64 = 2 - 2 \cdot 33 \equiv 2 \bmod \mathrm{N}.$$

Isto é,

$$\sqrt[\mathrm{E}]{29} = 2 = 29^d \bmod \mathrm{N}.$$

Para números primos diferentes p e q, denote

$$\mathrm{N} := pq \quad \text{e} \quad \phi(\mathrm{N}) := (p-1)(q-1).$$

Se

$$m = \mathrm{E}d \quad \text{e} \quad m \equiv 1 \bmod \phi(\mathrm{N}),$$

então

$$\sqrt[\mathrm{E}]{\cdot} \equiv \cdot^{d} \bmod \mathrm{N}.$$

Para quais E existe d tal que $\mathrm{E}d \equiv \bmod \phi(\mathrm{N})$?

Para todo E que é **relativamente primo** a $\phi(\mathrm{N})$, isto é, todo E que não tem nenhum divisor comum com $\phi(\mathrm{N})$.

Dado E relativamente primo a $\phi(\mathrm{N})$, como calcular tal d?

Pelo **Algoritmo de Euclides** (a seguir).

RSA = Rivest, Shamir e Adleman

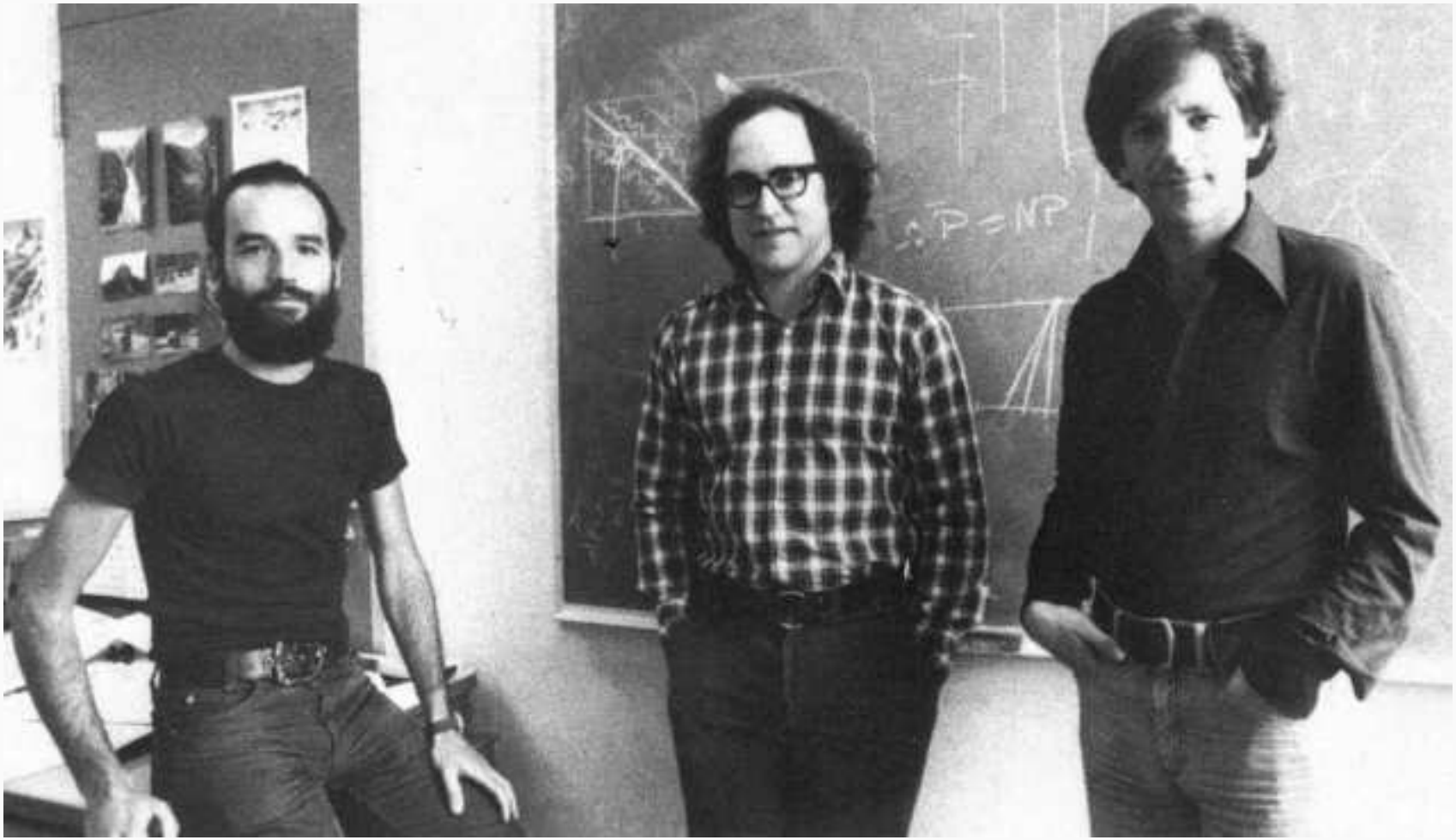


Figura 21: Os inventores do algoritmo RSA, Shamir, Rivest e Adleman

Passos no Algoritmo RSA

Para a Alice enviar a mensagem m secretamente ao Bob através de um canal inseguro,

1. Bob, para **gerar** a chave, escolhe
   - dois números primos p e q, e
   - um expoente E rel. primo a $\phi(\mathrm{N}) := (p-1)(q-1)$, e

   Bob, para **transmitir** a chave, envia à Alice
   - o produto $\mathrm{N} := pq$ e o expoente E.
2. Alice, para **cifrar**,
   - calcula $\mathrm{M} = m^{\mathrm{E}} \bmod \mathrm{N}$, e
   - transmite M ao Bob.
3. Bob, para **decifrar**,
   - calcula (por *Euclides*) d tal que $\mathrm{E}d \equiv 1 \bmod \phi(\mathrm{N})$, e
   - calcula $\mathrm{M}^d = m^{\mathrm{E}d} = m \bmod \mathrm{N}$ (pela *Fórmula de Euler*).

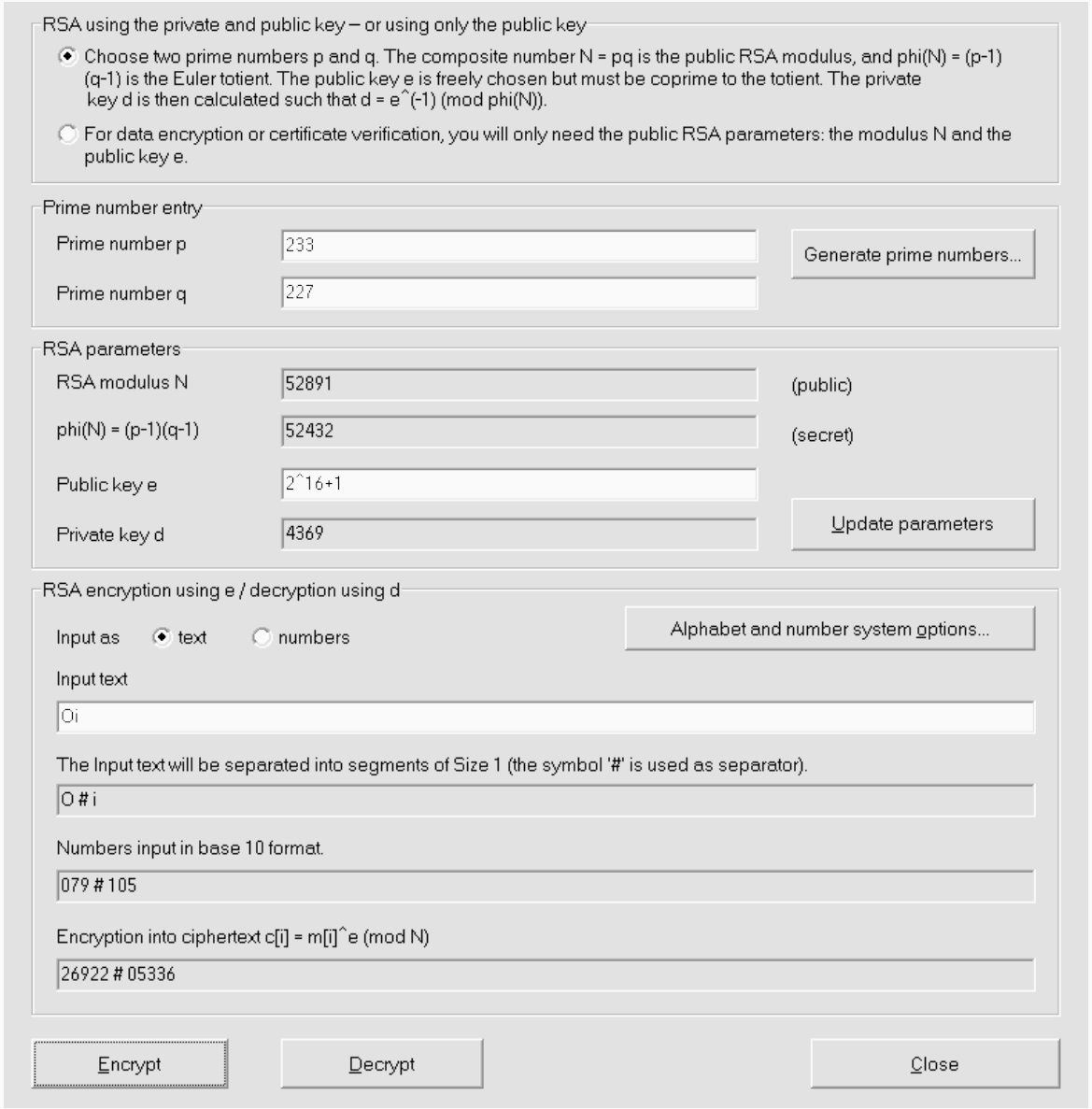


Figura 22: O CrypTool 1 oferece no Menu Individual Procedures -> RSA Cryptosystem a entrada RSA Demonstration para experimentar com os valores das chaves e da mensagem no RSA

Dificuldade = Computar a radiciação módulo N

A chave **pública** é E, e a **privada** é d. Como são **públicos**

- o módulo N,
- o expoente E e
- a mensagem cifrada M $(= m^{\mathrm{E}})$,

a segurança de RSA baseia-se na **dificuldade da computação da radiciação**

$$m \equiv \sqrt[\mathrm{E}]{\mathrm{M}}\ (= \mathrm{M}^{1/\mathrm{E}}) \bmod \mathrm{N}.$$

O **atalho** é o conhecimento dos fatores primos p e q de $\mathrm{N} = pq$ que possibilita calcular pelo *Algoritmo de Euclides*

- o módulo $\phi(\mathrm{N}) = (p-1)(q-1)$, e
- o inverso multiplicativo $d = 1/\mathrm{E}$ de E, isto é, d tal que

$$\mathrm{E}d \equiv 1 \bmod \phi(\mathrm{N});$$

para valer, pela *Fórmula de Euler*, $\mathrm{M}^d = m^{\mathrm{E}d} \equiv m \bmod \mathrm{N}$.

Dificuldade = Computar o Logaritmo módulo p

Um olheiro obteria a mensagem secreta m a partir de N, E e M, se pudesse computar

$$m \equiv \sqrt[\mathrm{E}]{\mathrm{M}}\ (= \mathrm{M}^{1/\mathrm{E}}) \bmod \mathrm{N}.$$

isto é, a *radiciação* $\sqrt[\mathrm{E}]{\cdot}$ que inverte a *potenciação* $x \mapsto x^d = y$,

$$\sqrt[\mathrm{E}]{y} = x \quad \text{com } x \text{ tal que } x^{\mathrm{E}} = y.$$

Enquanto a potenciação é facilmente computável, a **radiciação é dificilmente computável** para escolhas de p e q **suficientemente grande**:

O Teorema de Euclides garante que haja números primos arbitrariamente **grandes** (> 1024 bits),

$$\#\{ \text{números primos} \} = \infty$$

Acolchamento

Na prática, a mensagem m precisa de ser *enchida*, isto é, artificialmente aumentada.

Caso contrário, quando a mensagem clara m e o expoente E são pequenos (por exemplo, $\mathrm{E} = 3$), possivelmente a mensagem cifrada $\mathrm{M} = m^{\mathrm{E}}$ satisfaz $\mathrm{M} < \mathrm{N}$. Neste caso, a igualdade

$$m = \sqrt[\mathrm{E}]{\mathrm{M}} \quad \text{vale em } \mathbb{Z} (\text{e não só } \bmod \mathrm{N})!$$

Por isso, m é facilmente computável. Por exemplo,

- pelo **Método da Bisseção** já apresentada, ou,
- numericamente mais eficaz, pelo *método de Newton*.

Cifrar e Decifrar no RSA

Recordemo-nos de que no RSA são públicos o *produto* $\mathrm{N} = pq$ de dois números primos p e q, e a **chave pública** E para cifrar.

- Ciframos pela potenciação $\cdot^{\mathrm{E}}$, e
- deciframos pela raiz $\sqrt[\mathrm{E}]{\cdot} \bmod \mathrm{N}$.

Calcular a raiz $\sqrt[\mathrm{E}]{\cdot} \bmod \mathrm{N}$ é dificílimo; praticamente impossível!

Dado E relativamente primo a $\phi(\mathrm{N}) := (p-1)(q-1)$.

Se conhecemos p e q (e não só $\mathrm{N} = pq$), então conhecemos $\phi(\mathrm{N})$ e podemos calcular a **chave privada** d tal que

$$\mathrm{E} \cdot d \equiv 1 \bmod \phi(\mathrm{N}),$$

para computar $\cdot^{d} = \sqrt[\mathrm{E}]{\cdot} \bmod \mathrm{N}$! **Como calcular** d?

Resposta: Pelo **Algoritmo de Euclides**!

Divisores Comuns

Para dois números inteiros a e b,

$$\textbf{divisor comum} = \textbf{número inteiro positivo que divide } a \textbf{ e } b.$$

O **maior** divisor comum de a e b é o **maior** número inteiro positivo que divide a e b. Denote

$$\mathbf{mdc}(a, b) := \text{ o maior divisor comum de } a \text{ e } b \text{ .}$$

Exemplo para os dois números inteiros 24 e 36

$$\{ \text{ divisores comuns } \} = \{2, 3, 4, 6, \mathbf{12}\}$$

e

$$\text{mdc} = \text{ o maior divisor comum } = 12.$$

Computar o Maior Divisor Comum

Sejam a e b números inteiros positivos tal que $a \geq b$ e com

$$a = b \cdot q + r \quad \text{com } 0 \leq r < b. \qquad (\dagger)$$

Equação $(\dagger) \implies d \mid a, b$ se e tão-somente se $d \mid b, r$, $\implies$

$\{\text{divisores comuns de } a \text{ e } b\} = \{\text{divisores comuns de } b \text{ e } r\}$,

em particular

$$\mathbf{mdc}(a, b) = \mathbf{mdc}(b, r).$$

Reaplicando a divisão com resto aos números menores b e r,

$$b = r \cdot q' + r' \quad \text{com } 0 \leq r' < r$$

e por isto

$$\mathrm{mdc}(b, r) = \mathrm{mdc}(r, r').$$

Iterando, chegamos a $s := r'^{\cdots'}$ e $r'^{\cdots''}$ com $r'^{\cdots''} = 0$, isto é

$$\mathrm{mdc}(a, b) = \ldots = \mathbf{mdc}(s, 0) = \mathbf{s}.$$

Exemplo da Computação do Maior Divisor Comum pela iterada Divisão com Resto

Calculamos o maior divisor comum de $a = 748$ e $b = 528$ pela iterada divisão com resto:

$$\begin{aligned} 748 &= 528 \cdot 1 + 220 \\ 528 &= 220 \cdot 2 + 88 \\ 220 &= 88 \cdot 2 + \mathbf{44} \\ 88 &= 44 \cdot 2 + 0, \end{aligned}$$

logo $\mathrm{mdc}(528, 220) = 44$. Isto é,

o maior divisor comum $=$ o penúltimo resto .

O Matemático grego Euclides

Figura 23: Euclides (325 – 265 antes de Cristo)

Algoritmo de Euclides = Computação do Maior Divisor Comum pela *Iterada Divisão com Resto*

Algoritmo de Euclides

Sejam a e b números inteiros positivos com $a \geq b$. O seguinte algoritmo calcula $mdc(a, d)$ em um número finito de passos:

(**inicialização**) Ponha $r_0 = a$ e $r_1 = b$, e $i = 1$.
(**divisão com resto**) Divida r_{i-1} por r_i com resto para obter

$$r_{i-1} = r_i q_i + r_{i+1} \quad \text{com } 0 \leq r_{i+1} \leq r_i.$$

(**iteração**) Distinga entre:

- ou $r_{i+1} = 0$, então $r_i = \mathrm{mdc}(a, b)$ e o algoritmo termina,
- ou $r_{i+1} > 0$, então ponha $i := i + 1$ e continue no passo (*divisão com resto*).

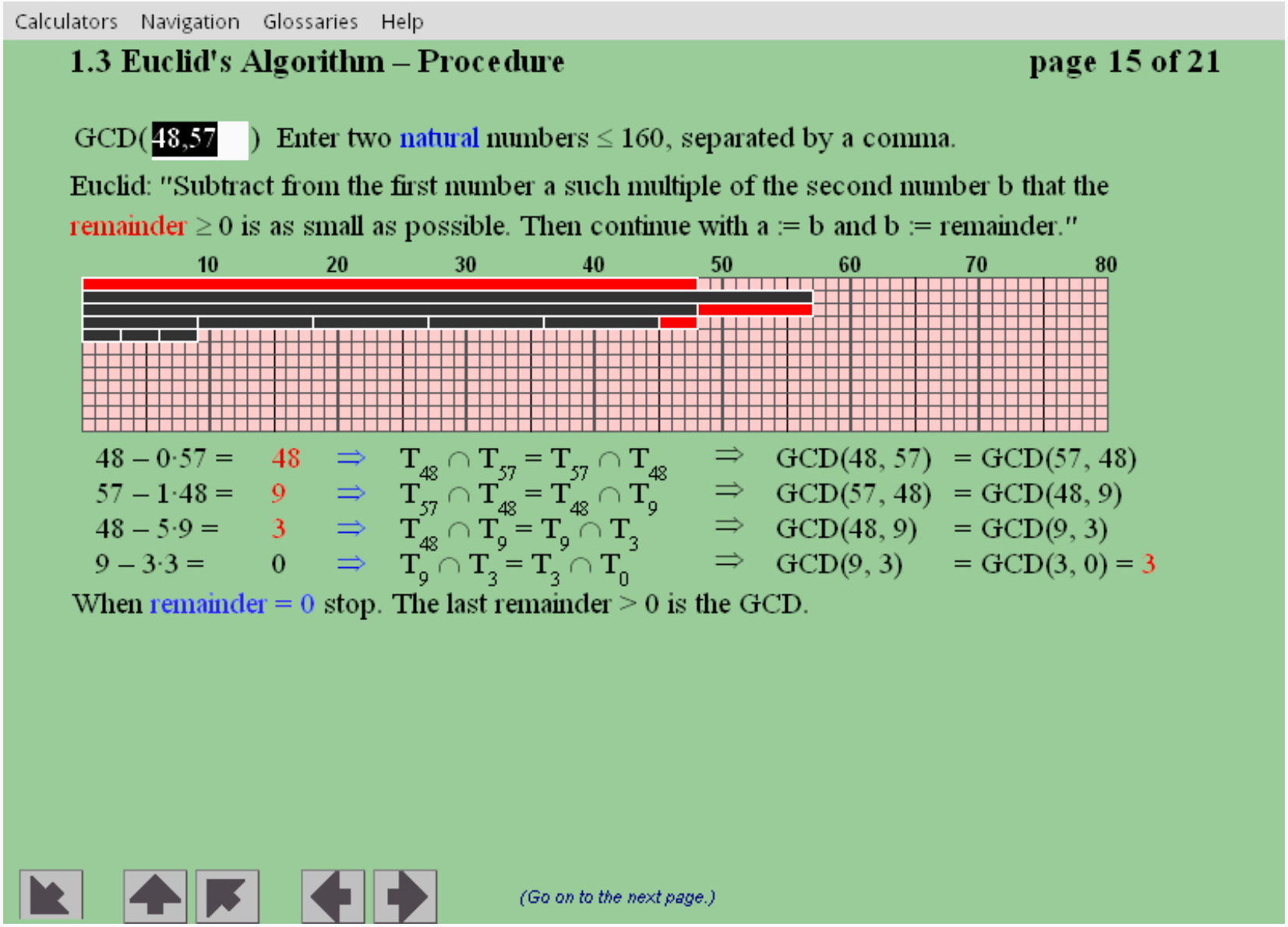


Figura 24: O `CrypTool 1`, na entrada do menu `Indiv. Procedures -> Number Theory Interactive -> Learning Tool for Number Theory`, Seção `1.3`, página 15, mostra uma animação deste algoritmo

Algoritmo de Euclides Estendido = Computar o Maior Divisor Comum como Soma de Múltiplos

Uma **soma de múltiplos** (ou *combinação linear*) de dois números inteiros a e b é uma soma

$$s = \lambda a + \mu b$$

para números inteiros λ e μ.

Exemplo. Para $a = 15$ e $b = 9$, uma soma de múltiplos deles é

$$s = 2 \cdot a + (-3) \cdot b = 2 \cdot 15 - 3 \cdot 9 = 3.$$

O *Algoritmo de Euclides Estendido* mostrará iterativamente que

maior divisor comum de a e b $=$ **soma de múltiplos** de a e b .

Isto é, há números inteiros u e v tais que

$$\mathrm{mdc}(a, b) = au + bv.$$

Maior Divisor Comum = Soma de Múltiplos

Inicialmente, com $r_0 := a$, $r_1 := b$,

$$r_0 = r_1 q_i + r_2 \quad \text{com } 0 \leq r_2 < r_1.$$

Isto é, $r_2 = r_0 - r_1 q_1$, $\Longrightarrow$

$$r_0, r_1, \text{ e } r_2 = \text{ somas de múltiplos de } a \text{ e } b \text{ .}$$

Por indução,

$$r_{i-1} = r_i q_i + r_{i+1} \quad \text{com } 0 \leq r_{i+1} \leq r_i.$$

Como r_{i-1} e r_i = somas de múltiplos de a e b, $\Longrightarrow$

$$r_{i+1} = r_{i-1} - r_i q_i = \text{ uma soma de múltiplos de } a \text{ e } b \text{ .}$$

Em particular, quando finalmente $r_{\mathrm{I}+1} = 0$, $\Longrightarrow$

$$r_{\mathrm{I}} = \mathrm{mdc}(r_{\mathrm{I}}, r_{\mathrm{I}+1}) = \mathrm{mdc}(a, b) = \text{ soma de múltiplos de } a \text{ e } b \text{ .}$$

Exemplo do Algoritmo de Euclides Estendido

Já calculamos o maior divisor comum de $a = 748$ e $b = 528$,

$$\begin{aligned} 748 &= 528 \cdot 1 + 220 \\ 528 &= 220 \cdot 2 + 88 \\ 220 &= 88 \cdot 2 + \mathbf{44} \\ 88 &= 44 \cdot 2 + 0, \end{aligned}$$

logo

$$\begin{aligned} 220 &= 748 - 528 \cdot 1 = a - b \\ 88 &= 528 - 220 \cdot 2 = b - (a-b)2 = 3b - 2a \\ \mathbf{44} &= 220 - 88 \cdot 2 = (a-b) - (3b-2a)2 = 5a - 7b, \end{aligned}$$

e, com efeito,

$$44 = 5 \cdot 748 - 7 \cdot 528.$$

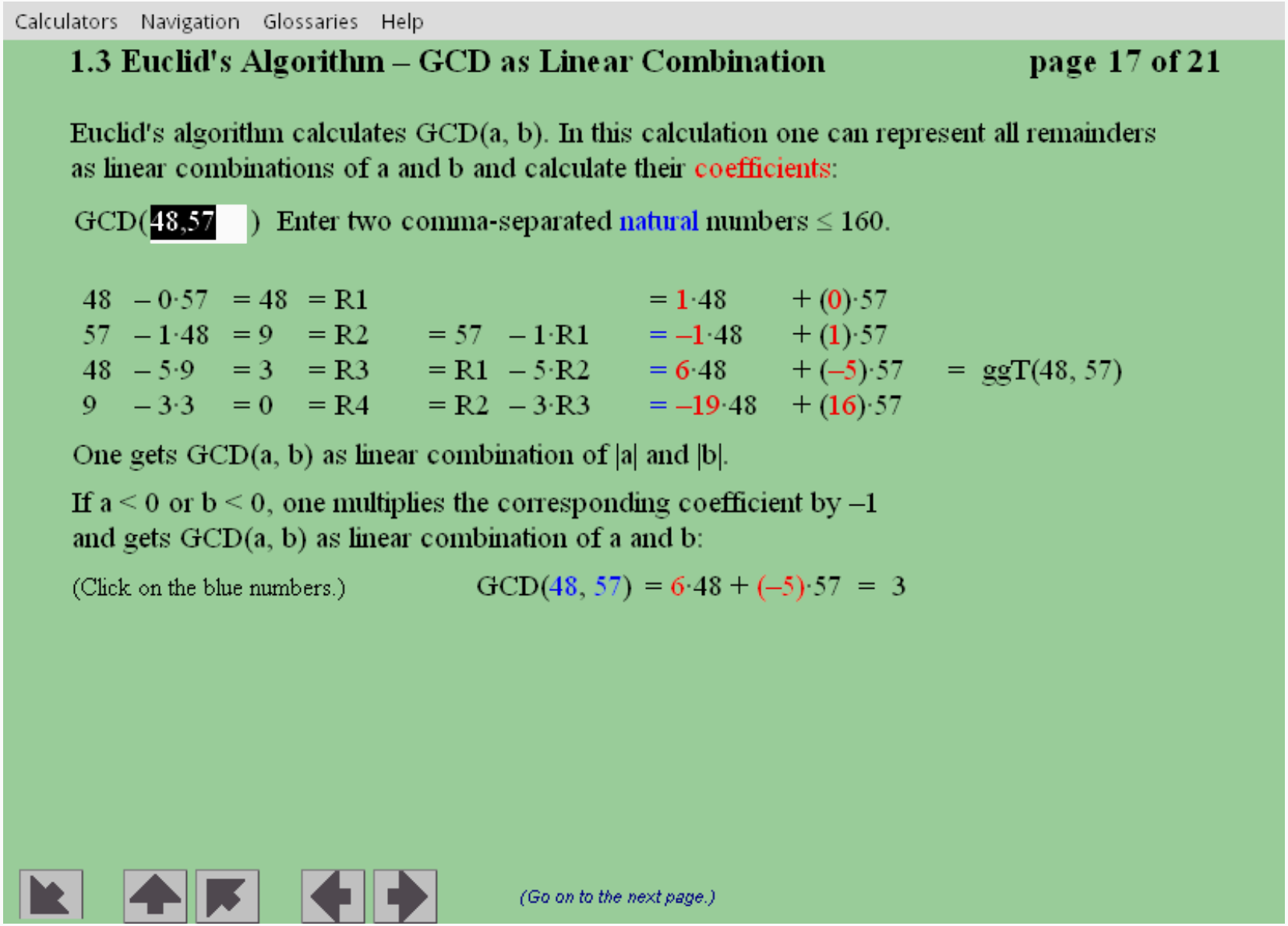


Figura 25: O CrypTool 1, na entrada do menu `Indiv. Procedures -> Number Theory Interactive -> Learning Tool for Number Theory`, Seção 1.3, página 17, mostra uma animação deste algoritmo.

Números Invertíveis

Em $\mathbb{Z}$, os únicos números que dividem todos são ± 1. Em $\mathbb{Z}/m\mathbb{Z}$, depende do módulo m com quais números podemos dividir,

- às vezes só pelos números invertíveis ± 1 (como em $\mathbb{Z}$), ou
- às vezes por todos os números (exceto 0).

Um número com que podemos dividir é uma *unidade* ou *número invertível*; caracterização mais exata:

Unidade

O elemento $\bar{x}$ em $\mathbb{Z}/m\mathbb{Z}$ é uma **unidade** se há $\bar{y}$ em $\mathbb{Z}/m\mathbb{Z}$ tal que

$$\bar{y}\bar{x} = 1.$$

*O elemento $\bar{y}$ é o **inverso** de $\bar{x}$ e denotado por $\bar{x}^{-1}$.* Denotamos

$$\mathbb{Z}/m\mathbb{Z}^* := \{ \text{as unidades em } \mathbb{Z}/m\mathbb{Z}\}.$$

Esta tabela de multiplicação revela as unidades $\mathbb{Z}/4\mathbb{Z}^* = \{1, 3\}$,

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

enquanto esta revela as unidades $\mathbb{Z}/5\mathbb{Z}^* = \{1, 2, 3, 4\}$:

*	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Caracterizar Unidades

Os números inteiros a e b são **relativamente primos** se

$$\mathrm{mdc}(a, b) = 1,$$

isto é, se nenhum número inteiro (fora 1) divide a e b. Por exemplo, são relativamente primos 28 e 45, mas 30 e 42 não.

Proposição (Caracterização de Unidades)

Dado um número inteiro x,

$\bar{x}$ é **unidade** em $\mathbb{Z}/m\mathbb{Z} \iff x$ e m são **relativamente primos**.

Exemplos

Por exemplo, $\mathrm{mdc}(1, 4) = \mathrm{mdc}(3, 4) = 1$, mas $\mathrm{mdc}(2, 4) = 2$,
$\implies$ (pela proposição) em $\mathbb{Z}/4\mathbb{Z}$ são unidades 1 e 3, mas 2 não.

Demonstração (da Caracterização de Unidades):

Pelo **Algoritmo de Euclides Estendido**, há u e v em $\mathbb{Z}$ tais que

$$ux + vm = \mathrm{mdc}(x, m).$$

Então $\mathrm{mdc}(x, m) = 1$ se e tão-somente se há u em $\mathbb{Z}$ tal que

$$ux \equiv 1 \bmod m$$

ou, equivalentemente,

$$\bar{u}\bar{x} = 1 \quad \text{em } \mathbb{Z}/m\mathbb{Z}.$$

Isto é, $\bar{x}$ é uma **unidade** em $\mathbb{Z}/m\mathbb{Z}$.

Corolário.

Se p é um número primo, então

$$(\mathbb{Z}/p\mathbb{Z})^* = \{1, ..., p-1\}.$$

Isto é, todos os elementos, exceto 0, são unidades.

Demonstração:

Se p é primo, então $\mathrm{mdc}(x, p) = 1$ para $x = 1, ..., p-1$.

O Corpo Finito de p Elementos

Um **corpo** é um anel cujos números são todos unidades ($\neq 0$). Isto é, um anel em que se pode dividir por qualquer número ($\neq 0$). Para um número primo p, pelo Corolário, $\mathbb{Z}/p\mathbb{Z}$ é um corpo, o **corpo com p elementos**, denotado por

$$\mathbf{F}_p := \mathbb{Z}/p\mathbb{Z}.$$

Teorema. *(Existência da **raiz primitiva** em um corpo finito)*

Se p é um número primo, então há α em $\mathbf{F}_p^* = \{1, 2, \ldots, p-1\}$ tal que

$$\mathbf{F}_p^* = \{\alpha, \alpha^2, \ldots\}.$$

Demonstração:

Como (pela iterada divisão com resto) um polinômio de grau m tem $\leq m$ raízes, e $x^m = 1$ se e tão-somente se $\mathrm{P}(x) = 0$ para $\mathrm{P}(\mathrm{X}) = \mathrm{X}^m - 1$, vale

$$\#\{x \text{ em } \mathbf{F}_p^* \text{ tal que } x^m = 1\} \leq m \quad (\dagger)$$

Mostramos que qualquer tal grupo G é *cíclico*, isto é, gerado por um elemento: Seja x um elemento em G de *ordem* (= o menor número $m > 0$ tal que $x^m = 1$) máxima. Se $m < \#\mathrm{G}$, então há por $(\dagger)$ um y em G cuja ordem não divide m. Então a ordem de $z = xy$ é $> m$, em *contradição à escolha* de m.

Anotações

Estão **disponíveis**

- os **eslaides** desta palestra e
- um **manuscrito** sobre criptografia que aprofunda o que aprendemos

online em `konfekt.bitbucket.io/talks/criptografia`

Referências

Diffie, Whitfield, e Martin Hellman. 1976. «New directions in cryptography». *IEEE transactions on Information Theory* 22 (6). IEEE: 644–54.

Rivest, Ronald L, Adi Shamir, e Leonard Adleman. 1978. «A method for obtaining digital signatures and public-key cryptosystems». *Communications of the ACM* 21 (2). ACM: 120–26.

1. Aritmética Modular
2. Detectar Primos
3. Troca de Chaves
4. O Algoritmo RSA
5. Algoritmo de Euclides