

# Criptografia e Curvas Elípticas

Thiago Holleben

Universidade Federal do Rio de Janeiro

*hollebenthiago@gmail.com*

9 de junho de 2022

- História
- Criptografia
- ECC
- Exemplos
- Parâmetros
- Contando pontos
- Primalidade

# História (Neal Koblitz e Victor Miller - ECC, 1985)



# História (Hendrik Lenstra - Primalidade, 1984)



# História (René Schoof - Contagem de pontos, 1985)



# Criptografia simétrica



# Criptografia simétrica - Alguns problemas

- As duas entidades precisam ter acesso à uma mesma chave que não pode ser pública

# Criptografia simétrica - Alguns problemas

- As duas entidades precisam ter acesso à uma mesma chave que não pode ser pública
- Para situações reais como por exemplo comunicação entre um banco e seus clientes, se torna inviável devido a quantidade de troca de chaves necessárias

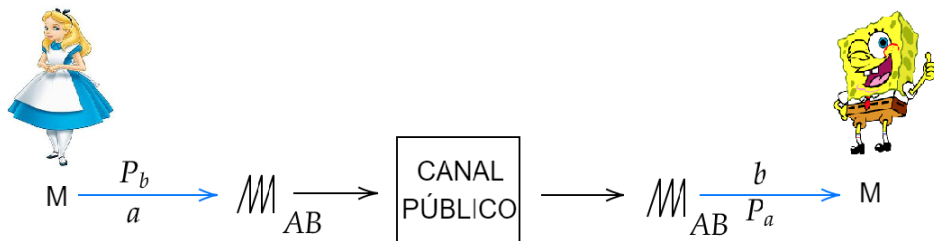


# Uma alternativa (criptografia assimétrica)

Se cada pessoa possui um par de chaves: uma pública e uma privada, os dois problemas são resolvidos:

# Uma alternativa (criptografia assimétrica)

Se cada pessoa possui um par de chaves: uma pública e uma privada, os dois problemas são resolvidos:



Vamos tentar entender melhor alguns exemplos de funções que representam as setas azuis na imagem acima. Normalmente elas são chamadas de *one way functions* ou então *trapdoor functions*

# One way functions

Dizemos que uma função  $f: A \rightarrow B$  é uma *one way function* se:

- É "fácil" calcular  $f(a)$

# One way functions

Dizemos que uma função  $f: A \rightarrow B$  é uma *one way function* se:

- É "fácil" calcular  $f(a)$
- Dado  $f(x)$  é "difícil" calcular  $x$ , mas existe uma informação  $t$  tal que, sabendo  $t$  e  $f(x)$  é "fácil" calcular  $x$

# One way functions

Dizemos que uma função  $f: A \rightarrow B$  é uma *one way function* se:

- É "fácil" calcular  $f(a)$
- Dado  $f(x)$  é "difícil" calcular  $x$ , mas existe uma informação  $t$  tal que, sabendo  $t$  e  $f(x)$  é "fácil" calcular  $x$

Para definir formalmente o que significa "fácil" e "difícil" precisaríamos de noções da teoria de complexidade computacional da ciência da computação

# One way functions

Dizemos que uma função  $f: A \rightarrow B$  é uma *one way function* se:

- É "fácil" calcular  $f(a)$
- Dado  $f(x)$  é "difícil" calcular  $x$ , mas existe uma informação  $t$  tal que, sabendo  $t$  e  $f(x)$  é "fácil" calcular  $x$

Para definir formalmente o que significa "fácil" e "difícil" precisaríamos de noções da teoria de complexidade computacional da ciência da computação

Mesmo essas funções sendo a base de grande parte da criptografia que utilizamos hoje em dia, não sabemos se alguma função satisfazendo as duas condições acima de fato existe

# Não tem problema!

Mesmo não sendo provada a existência de *one way functions*, temos boas candidatas. Além disso, uma prova não construtiva de que alguma das funções que são utilizadas hoje em dia não é uma *one way function* não é suficiente

# Não tem problema!

Mesmo não sendo provada a existência de *one way functions*, temos boas candidatas. Além disso, uma prova não construtiva de que alguma das funções que são utilizadas hoje em dia não é uma *one way function* não é suficiente

*One way functions* são baseadas em problemas matemáticos, alguns exemplos:

- Fatoração de números inteiros (RSA)
- Logaritmo discreto



# Não tem problema!

Mesmo não sendo provada a existência de *one way functions*, temos boas candidatas. Além disso, uma prova não construtiva de que alguma das funções que são utilizadas hoje em dia não é uma *one way function* não é suficiente

*One way functions* são baseadas em problemas matemáticos, alguns exemplos:

- Fatoração de números inteiros (RSA)
- Logaritmo discreto

ECC é baseada no problema do logaritmo discreto como veremos mais para frente

## Definição

Dados  $g, h \in \mathbb{N}$  não nulos e um número  $n$ , queremos encontrar o menor valor  $a \in \mathbb{N}$  tal que  $g^a = h \pmod{n}$

## Definição

Dados  $g, h \in \mathbb{N}$  não nulos e um número  $n$ , queremos encontrar o menor valor  $a \in \mathbb{N}$  tal que  $g^a = h \pmod{n}$

A dificuldade de encontrar o logaritmo discreto depende bastante dos parâmetros  $g, h, n$ .

# Troca de chaves Diffie-Hellman

Suponha que Alice e Bob querem se comunicar porém o único meio de comunicação entre eles é um canal público não confiável. Alice e Bob, usando o canal público concordam em usar um certo  $n$ , um elemento  $g \neq 0, 1$ .

# Troca de chaves Diffie-Hellman

Suponha que Alice e Bob querem se comunicar porém o único meio de comunicação entre eles é um canal público não confiável. Alice e Bob, usando o canal público concordam em usar um certo  $n$ , um elemento  $g \neq 0, 1$ .

Alice e Bob "escolhem" números naturais  $a, b$  tais que  $a, b < |\{g^0, g^1, \dots\}|$ .

Após a escolha ser feita, Alice e Bob calculam respectivamente  $P_a = g^a$  e  $P_b = g^b$  e divulgam no canal público apenas  $P_a$  e  $P_b$

# Troca de chaves Diffie-Hellman

Suponha que Alice e Bob querem se comunicar porém o único meio de comunicação entre eles é um canal público não confiável. Alice e Bob, usando o canal público concordam em usar um certo  $n$ , um elemento  $g \neq 0, 1$ .

Alice e Bob "escolhem" números naturais  $a, b$  tais que  $a, b < |\{g^0, g^1, \dots\}|$ .

Após a escolha ser feita, Alice e Bob calculam respectivamente  $P_a = g^a$  e  $P_b = g^b$  e divulgam no canal público apenas  $P_a$  e  $P_b$

Chamamos os números naturais  $a$  e  $b$  de *chaves privadas* e os elementos do grupo  $P_a$  e  $P_b$  de *chaves públicas*

# Troca de chaves Diffie-Hellman

Suponha que Alice e Bob querem se comunicar porém o único meio de comunicação entre eles é um canal público não confiável. Alice e Bob, usando o canal público concordam em usar um certo  $n$ , um elemento  $g \neq 0, 1$ .



Alice e Bob "escolhem" números naturais  $a, b$  tais que  $a, b < |\{g^0, g^1, \dots\}|$ .

Após a escolha ser feita, Alice e Bob calculam respectivamente  $P_a = g^a$  e  $P_b = g^b$  e divulgam no canal público apenas  $P_a$  e  $P_b$

Chamamos os números naturais  $a$  e  $b$  de *chaves privadas* e os elementos do grupo  $P_a$  e  $P_b$  de *chaves públicas*

Note que  $P_a^b = g^{ab} = g^{ba} = P_b^a$ , e como é "difícil" descobrir  $a$  ou  $b$  dado  $P_a$  ou  $P_b$ , o elemento do grupo  $P_a^b$  é um segredo entre Alice e Bob

# Troca de chaves Diffie-Hellman

	CANAL PÚBLICO	
$G$	$G$	$G$
$g$	$g$	$g$
$a$		
$P_a$	$P_a$	$P_a$
		$b$
$P_b$	$P_b$	$P_b$
$P_a^b$		$P_b^a$



# Curvas elípticas sobre $\mathbb{R}$

Considere o polinômio em 2 variáveis  $x, y$ :

$$E : y^2 = x^3 + ax + b, \quad a, b \in \mathbb{R}$$

Denotamos o conjunto de pontos  $\{(x, y) \in \mathbb{R}^2 \mid y^2 = x^3 + ax + b\}$  os pontos (fora do infinito) da curva elíptica  $E$ . Abusando (bastante!) da notação, vamos assumir que o "ponto"  $(\infty, \infty)$  também é um ponto da curva.

# Somando pontos e curvas módulo $p$

Podemos considerar as mesmas hipóteses do slide anterior mas considerando as equações mod  $p$  (primo):

$$E : y^2 = x^3 + ax + b \bmod p, \quad a, b < p$$

Novamente, o importante é o conjunto de pontos:

$$\{(\infty, \infty)\} \cup \{(x, y) \in [0, \dots, p-1] \times [0, \dots, p-1] \mid y^2 = x^3 + ax + b \bmod p\}$$

# Elliptic Curve Diffie-Hellman (ECDH)

Sabemos que a equação de uma curva elíptica  $E$  módulo  $p$  com  $p \neq 2, 3$  é:

$$y^2 = x^3 + ax + b \pmod{p}$$

Com  $a, b < p$ .

# Elliptic Curve Diffie-Hellman (ECDH)

Sabemos que a equação de uma curva elíptica  $E$  módulo  $p$  com  $p \neq 2, 3$  é:

$$y^2 = x^3 + ax + b \pmod{p}$$

Com  $a, b < p$ .

Portanto, para poder ocorrer a troca de chaves, Alice e Bob precisam concordar nos seguintes parâmetros:

- Um primo  $p$
- Dois números menores que  $p$ :  $a$  e  $b$
- Um ponto  $P$  (fora do infinito) de  $E$

Note que ao concordarem em utilizar o ponto  $P$ , Alice e Bob também concordam na ordem  $n$  do ponto  $P$ , isto é,  $n = |\langle P \rangle|$

Já sabemos como Alice e Bob podem ter um segredo em comum, vamos ver agora brevemente como podemos transformar uma mensagem (em texto) em um ponto de uma curva elíptica concordada entre Alice e Bob

Já sabemos como Alice e Bob podem ter um segredo em comum, vamos ver agora brevemente como podemos transformar uma mensagem (em texto) em um ponto de uma curva elíptica concordada entre Alice e Bob

Suponha que Alice e Bob concordaram em um primo  $p$ , na curva  $E: y^2 = x^3 + ax + b \bmod p$  e em um ponto  $P \in E$  fora do infinito

Todo símbolo (caractere) que utilizamos ao digitar uma mensagem pode ser representado como um número natural entre 0 e 255, de forma que dada uma string  $m_0m_1 \dots m_k$ , podemos representá-la como o número:

$$P_{m_x} = o(m_0)256^0 + \dots o(m_k)256^k$$

Onde  $o(m_i)$  é o número correspondente ao caractere  $m_i$

Todo símbolo (caractere) que utilizamos ao digitar uma mensagem pode ser representado como um número natural entre 0 e 255, de forma que dada uma string  $m_0m_1 \dots m_k$ , podemos representá-la como o número:

$$P_{m_x} = o(m_0)256^0 + \dots o(m_k)256^k$$

Onde  $o(m_i)$  é o número correspondente ao caractere  $m_i$

- Nossa ideia vai ser considerar um ponto  $P_m$  com uma coordenada  $x$  próxima de  $P_{m_x}$ , de forma que podemos assumir que de fato existe um ponto em  $E$  cuja coordenada  $x$  é  $P_{m_x}$



Todo símbolo (caractere) que utilizamos ao digitar uma mensagem pode ser representado como um número natural entre 0 e 255, de forma que dada uma string  $m_0m_1 \dots m_k$ , podemos representá-la como o número:

$$P_{m_x} = o(m_0)256^0 + \dots o(m_k)256^k$$

Onde  $o(m_i)$  é o número correspondente ao caractere  $m_i$

- Nossa ideia vai ser considerar um ponto  $P_m$  com uma coordenada  $x$  próxima de  $P_{m_x}$ , de forma que podemos assumir que de fato existe um ponto em  $E$  cuja coordenada  $x$  é  $P_{m_x}$
- Além disso, precisamos que  $P_{m_x} < p$  e portanto o primo impõe uma restrição do tamanho da mensagem. Na prática, se a mensagem for grande demais, basta quebrar a mensagem em mensagens menores

Temos então um método de encriptar uma mensagem de texto qualquer em um ponto da curva elíptica  $E$  combinada entre Alice e Bob  
Podemos finalmente descrever como Alice pode mandar uma mensagem "segura" para Bob:

Temos então um método de encriptar uma mensagem de texto qualquer em um ponto da curva elíptica  $E$  combinada entre Alice e Bob  
Podemos finalmente descrever como Alice pode mandar uma mensagem "segura" para Bob:

- Alice e Bob "decidem" suas chaves privadas  $n_a$  e  $n_b$  respectivamente e com elas calculam suas chaves públicas  $n_aP$  e  $n_bP$  respectivamente

Temos então um método de encriptar uma mensagem de texto qualquer em um ponto da curva elíptica  $E$  combinada entre Alice e Bob

Podemos finalmente descrever como Alice pode mandar uma mensagem "segura" para Bob:

- Alice e Bob "decidem" suas chaves privadas  $n_a$  e  $n_b$  respectivamente e com elas calculam suas chaves públicas  $n_aP$  e  $n_bP$  respectivamente
- Alice transforma sua mensagem  $m$  em um ponto  $P_m$  cuja coordenada  $x$  é  $P_{m_x}$

Temos então um método de encriptar uma mensagem de texto qualquer em um ponto da curva elíptica  $E$  combinada entre Alice e Bob

Podemos finalmente descrever como Alice pode mandar uma mensagem "segura" para Bob:

- Alice e Bob "decidem" suas chaves privadas  $n_a$  e  $n_b$  respectivamente e com elas calculam suas chaves públicas  $n_aP$  e  $n_bP$  respectivamente
- Alice transforma sua mensagem  $m$  em um ponto  $P_m$  cuja coordenada  $x$  é  $P_{m_x}$
- Alice gera um número natural aleatório  $t$ , calcula os pontos  $t(n_aP)$  e  $P_m + t n_a(n_bP)$  e os envia para Bob

E por fim, Bob deve realizar os seguintes passos para recuperar a mensagem original

E por fim, Bob deve realizar os seguintes passos para recuperar a mensagem original

- Bob multiplica o primeiro ponto enviado por Alice por sua chave privada  $n_b$ . Bob agora possui os pontos  $tn_an_bP$  e  $P_m + tn_an_bP$

E por fim, Bob deve realizar os seguintes passos para recuperar a mensagem original

- Bob multiplica o primeiro ponto enviado por Alice por sua chave privada  $n_b$ . Bob agora possui os pontos  $tn_an_bP$  e  $P_m + tn_an_bP$
- Bob subtrai o segundo ponto pelo primeiro de forma que Bob agora conhece o ponto  $P_m$



E por fim, Bob deve realizar os seguintes passos para recuperar a mensagem original

- Bob multiplica o primeiro ponto enviado por Alice por sua chave privada  $n_b$ . Bob agora possui os pontos  $tn_an_bP$  e  $P_m + tn_an_bP$
- Bob subtrai o segundo ponto pelo primeiro de forma que Bob agora conhece o ponto  $P_m$
- Bob escreve a coordenada  $x$  de  $P_m$  na base 256 recuperando os números naturais  $o(m_i)$ , e portanto Bob finalmente descobre a mensagem de Alice

# Exemplo

Parâmetros públicos:

# Exemplo

Parâmetros públicos:

- $p = 2^{521} - 1$

Parâmetros públicos:

- $p = 2^{521} - 1$
- $E: y^2 = x^3 - 3x + 1093849038073734274511112390766805569936207598951683748994586394495953116150735016013708737573759623248592132296706313309438452531591012912142327488478985984$
- $|E| = 6864797660130609714981900799081393217269435300143305409394463459185543183397655394245057746333217197532963996371363321113864768612440380340372808892707005449$
- $P = (6021620849719266644313485397659641177743154559432487051941413478692850955441702394729964425257594712287487500155649468348602990174318433816864442395311520445 : 1584952542993235804449382063457676361099748412937303436394851183459258487460698729485745314661375236972159812294476280869907176470135877187668739352726297379 : 1)$

## Chaves públicas:

- Alice: (278089309606698148315420825277634377839118643324022871609229207984598412877999253050104215771833587797838337852299162804865596860676895387110362258197500824 : 4029761634265465356718296801583301361099214623320319354994766227452823584753975741510719205845830322937208067741113155363603504724145465042871974645335394073 : 1)
- Bob: (2898552068342607411354011383439396588315164396328066893332061793087203567620681712105961477494503768865213489065657780698904016456065090178904479782913363501 : 2745906793591004927708883303991980750407037732778937413643511963222701535921665393255729305100297915673624477872076548580005553288121641439170339318033776132 : 1)

Chaves privadas:

- Alice: 39146454526826757462569054833624462481238657697730531  
1543154112437898353989757891721105148729007966936811696973  
1671738172987377321947918685990842151669803465
- Bob: 603274360521577465669348407027522186109954987031329578  
0736453489626992849809832960755537613774984513220749525479  
316193630261819474852778151057962194293268778

# Exemplo

- Mensagem de Alice para Bob: "oi bob tudo bom?"
- Primeiro ponto da mensagem encriptada: (535566995305838843569339753042045682419713453127020598181197886635115545883956878333239266139529727602467565226759770888501010645838332801545742306089388574 : 1267832422498685711067624726639330233571789233932419007426212512450782204817020095490256297322477106904629543090172848746115681092878917243941864626456714482 : 1)
- Segundo ponto da mensagem encriptada: (332983521077068492740551374505846833163162148078609650069794887241999042831958918255881738632229680701613257878915592269657237222061287529146322692674324115 : 4431990014576051413212961156551205569391381549496899807905237462288784275322061290054583918113217030495975821601194222831762521356579659372995120064367526458 : 1)
- Mensagem que Bob vê após deciptar: "oi bob tudo bom?"

# Um exemplo beeeem mais interessante 1



# Um exemplo beeeem mais interessante 1

Original



Decrypted with right private key



Decrypted with wrong private key 1

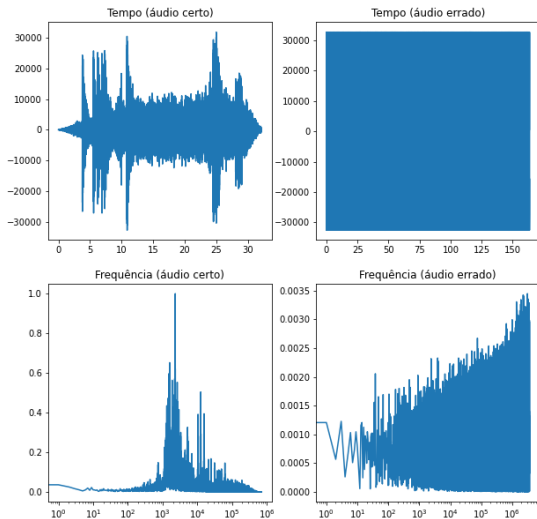


Decrypted with wrong private key 2



# Um exemplo beeeem mais interessante 2

# Um exemplo beeeem mais interessante 2



# Man in the middle attack (MITM Attack)

Suponha que Alice quer mandar uma mensagem para Bob, mas desta vez, vamos assumir também que existe uma outra entidade Eve que tem o poder de alterar mensagens que passam pelo canal público

Para ocorrer a troca de mensagens entre Alice e Bob, Alice precisa saber a chave pública de Bob. Se Eve quiser interferir na comunicação entre Alice e Bob, Eve pode guardar a chave pública de Bob e enviar a sua própria chave pública para Alice.

Desta forma, Eve consegue ler, encriptar e decriptar mensagens de Alice para Bob. Para evitar este ataque é preciso de uma *assinatura* nas mensagens que comprovam o remetente da mensagem.

Para aumentar a segurança de uma mensagem encriptada com *RSA* ou *ECC*, aumenta-se o tamanho das chaves privadas, do primo, etc. Aumentando o tamanho das chaves, a fatoração de números inteiros e o logaritmo discreto ficam mais "difíceis"

# ECC vs RSA

Para aumentar a segurança de uma mensagem encriptada com *RSA* ou *ECC*, aumenta-se o tamanho das chaves privadas, do primo, etc. Aumentando o tamanho das chaves, a fatoração de números inteiros e o logaritmo discreto ficam mais "difíceis"

Tamanho das chaves (em <i>bits</i> ) <i>ECC</i>	Tamanho das chaves (em <i>bits</i> ) <i>RSA</i>
163	1024
256	3072
384	7680
512	15360

Isso ocorre pois existem ataques em tempo subexponencial bons contra fatoração de inteiros (*RSA*), mas não existem ataques em tempo subexponencial que funcionem para qualquer curva elíptica

# De volta para curvas elípticas

Como foi mencionado anteriormente, para ocorrer a troca de mensagens entre Alice e Bob, é necessário saber a ordem de um ponto  $P$  na curva elíptica concordada  $E$ .



# De volta para curvas elípticas

Como foi mencionado anteriormente, para ocorrer a troca de mensagens entre Alice e Bob, é necessário saber a ordem de um ponto  $P$  na curva elíptica concordada  $E$ .

Pelo problema do logaritmo discreto, vimos que é importante saber a estrutura dos pontos de  $E$ .

## Distribuição - Exemplo 137

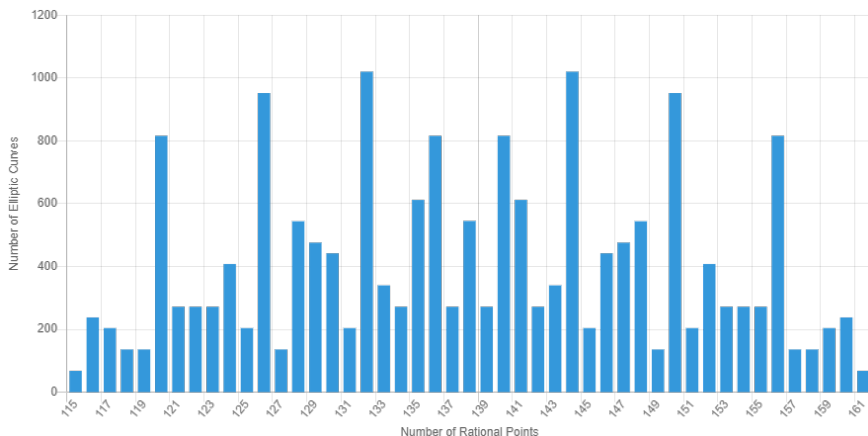
É possível provar que para qualquer curva elíptica  $E$  módulo  $p$ ,  $N = |E|$  satisfaz:

$$|N - (p + 1)| \leq 2\sqrt{p}$$

# Distribuição - Exemplo 137

É possível provar que para qualquer curva elíptica  $E$  módulo  $p$ ,  $N = |E|$  satisfaz:

$$|N - (p + 1)| \leq 2\sqrt{p}$$



# Outra aplicação - Teste de primalidade

## Outra aplicação - Teste de primalidade

Inputs: Um inteiro  $N$ , um limite  $L$  e um número máximo de curvas  $n$

Output: Um fator primo de  $N$

**for**  $i = 0, 1, \dots, n$  **do**

Escolha um ponto aleatório  $P = (x, y)$  tal que  $x, y \in \frac{\mathbb{Z}}{N\mathbb{Z}}$

Escolha um elemento aleatório  $A$  de  $\frac{\mathbb{Z}}{N\mathbb{Z}}$

Defina  $B = y^2 - x^3 - Ax$

Multiplicação escalar será na curva  $E: y^2 = x^3 + Ax + B$

**for**  $j = 1, 2, \dots, L$  **do**




$P = jP$

Caso não seja possível continuar a conta pois um elemento

$t \in \frac{\mathbb{Z}}{N\mathbb{Z}}$  não tem inverso,  $\gcd(t, N) > 1$  provavelmente é um fator não trivial de  $N$

**end for**

**end for**

-  SILVERMAN. The Arithmetic of Elliptic Curves 2. ed. Springer, 2009
-  STICHTENOTH. Function fields and algebraic codes 1. ed. Springer, 1993
-  WASHINGTON. Elliptic Curves Number Theory and Cryptography 2. ed. Taylor & Francis Group, 2008