Segurança da Informacção
Aula ECC

Ivan Sendin

FACOM - Universidade Federal de Uberlandia
ivansendin@yahoo.com,sendin@ufu.br

11 de junho de 2024

- ECC
- Elliptic Curve Cryptography
- **Ciframento Assimétrico** e **Assinaturas**
- Chaves de 256 bits
  Concorrente: RSA com 4k bits
- Qualquer sequencia de bits...
- Aula: introdução **rápida** sobre curvas e seus usos em criptografia...
- MUITAS OMISSÕES

- Ciframento Simétrico é o “normal”...
- ...faz parte da nossa “cultura”
- Uso uma chave/segredo para embaralhar uma informação
- Transmito, armazeno,....
- (Eu ou outra pessoa) Uso a chave para desembaralhar a informação
- $\mathcal{D}_K(\mathcal{E}_K(x)) = x$
- $\mathcal{D}_{k1}(\mathcal{E}_{k2}(x)) \neq x$ se $k1 \neq k2$

- Ainda: ECC é (uma) base para criptografia pós-quântica
  Eva tem acesso a um computador quântico...voce não!
- ciframento homomorfico

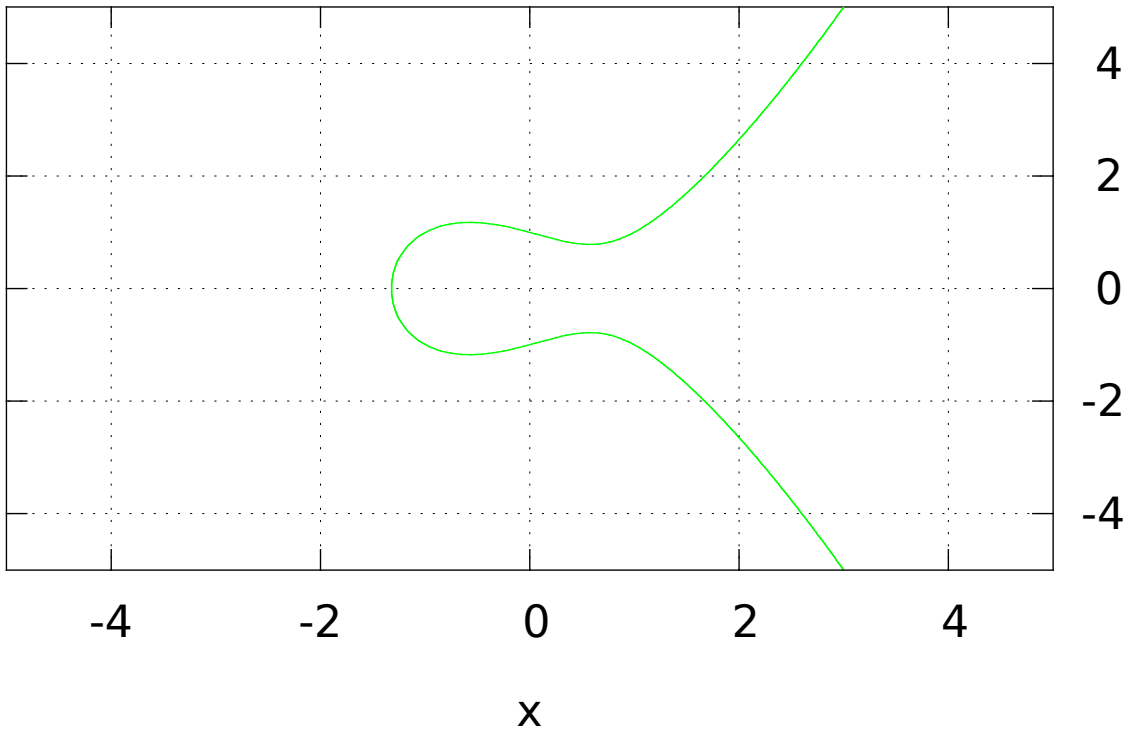
$$E(x) + E(y) == E(x + y)$$

- e ZK-Snarks

Uma curva é definida pela formula:

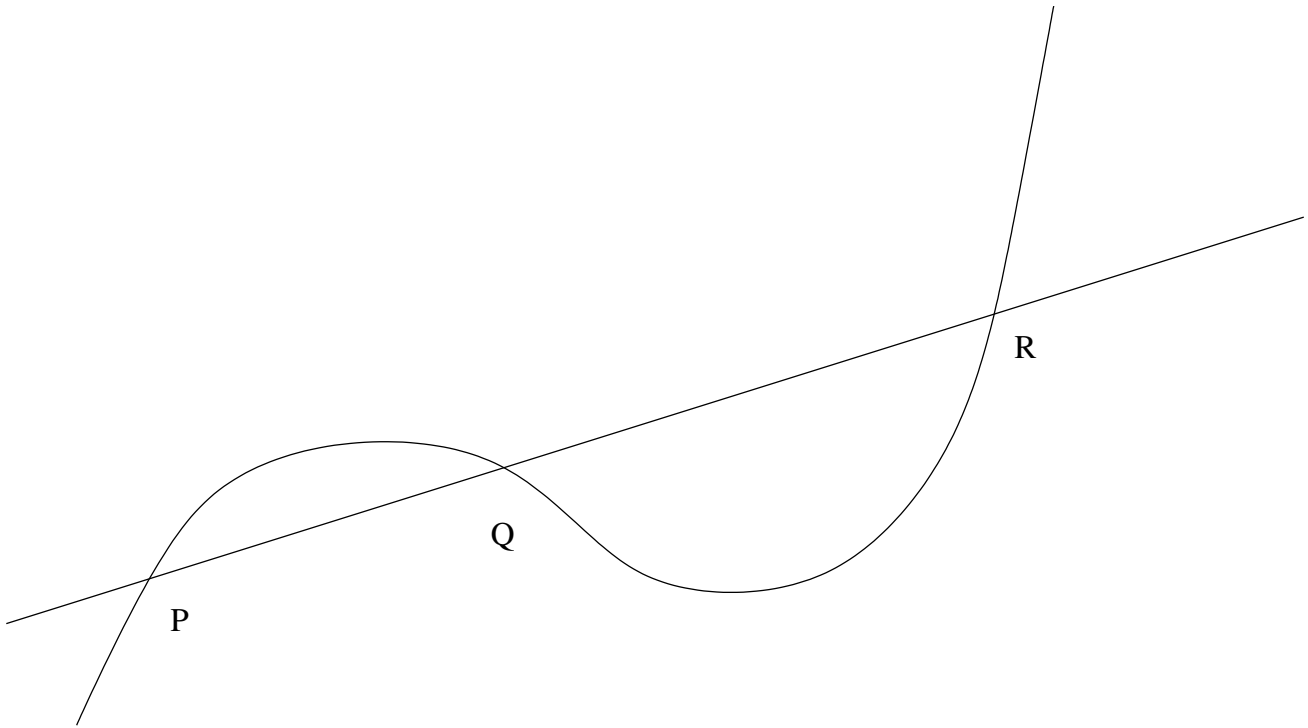
$$y^2 = x^3 + ax + b$$

- a e b sao os coeficientes de caracteristica da curva
- Para $a = -1$ e $b = 1$...


4
2
0
-2
-4
-4
-2
0
2
4
x

Soma

- Vamos definir a soma sobre pontos de uma curva...
- **Inventar**
- $R = P + Q$
- "Linha reta passando por P e Q que chega em R"
- $R = (1, 2) + (3, 5)$ (para $a = ...$)

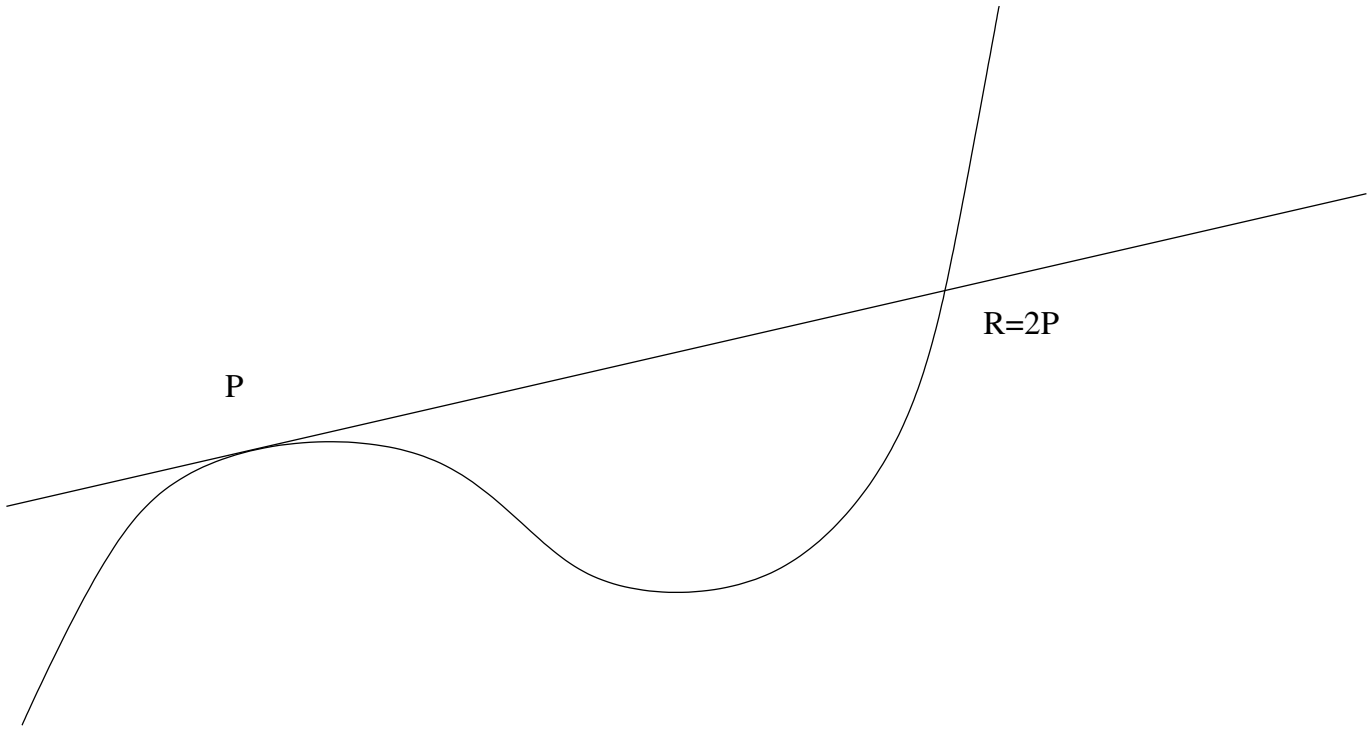

R
Q
P

Soma

- $R = P + Q$
- Calcula a formula da reta dado P e Q
- Junta com a fórmula da curva....
- Temos um método para somar pontos

Duplicação

- Se $P == Q$ a soma é uma duplicação
- (caso special...)
- $R = P + P$
- Tangente da curva...


R=2P
P

Multiplicação por escalar

- Sei calcular $P + Q$ e $2P$
- Dado P, como calcular, por exemplo $17P$??

Multiplicação por escalar

- $17P$???
- Calculo $(((P + P) + P) + P...)$
- Custo disso???
- Ou ...

Multiplicação por escalar

- $17P$???
- Calculo $(((P + P) + P) + P...)$
- Ou
- $P \rightarrow 2P \rightarrow 4P \rightarrow 8P \rightarrow 16P$
- $17P = P + 16P$
- (Questão de prova: como calcular $177P$ de forma eficiente?)

Divisão

- Dado P
- Dado $Q = nP$
- Como "dividir" Q por P e determinar n?
- (O nome correto é calcular logaritmo...)

```
func achaN(P,Q): #Q=nP
      n=1
      T=P
      Enquanto T!=Q:
            T=T+P
            n=n+1
      return n
```

Divisão

- Q = 17P = 2(2(2(2(P)))) + P - 5 operações
- "Divisão"
- 2P=Q? 3P=Q? 4P=Q?.... 17 operações
- Para n grande a divisão não pode ser executada
- (antes do universo acabar...)

Divisão

- Para uma curva conhecida (a,b)
- Para um determinado Q e outro ponto P
- Eu posso "colocar"o ponto na curva e ter - pelo menos - alguma ideia do valor de n...

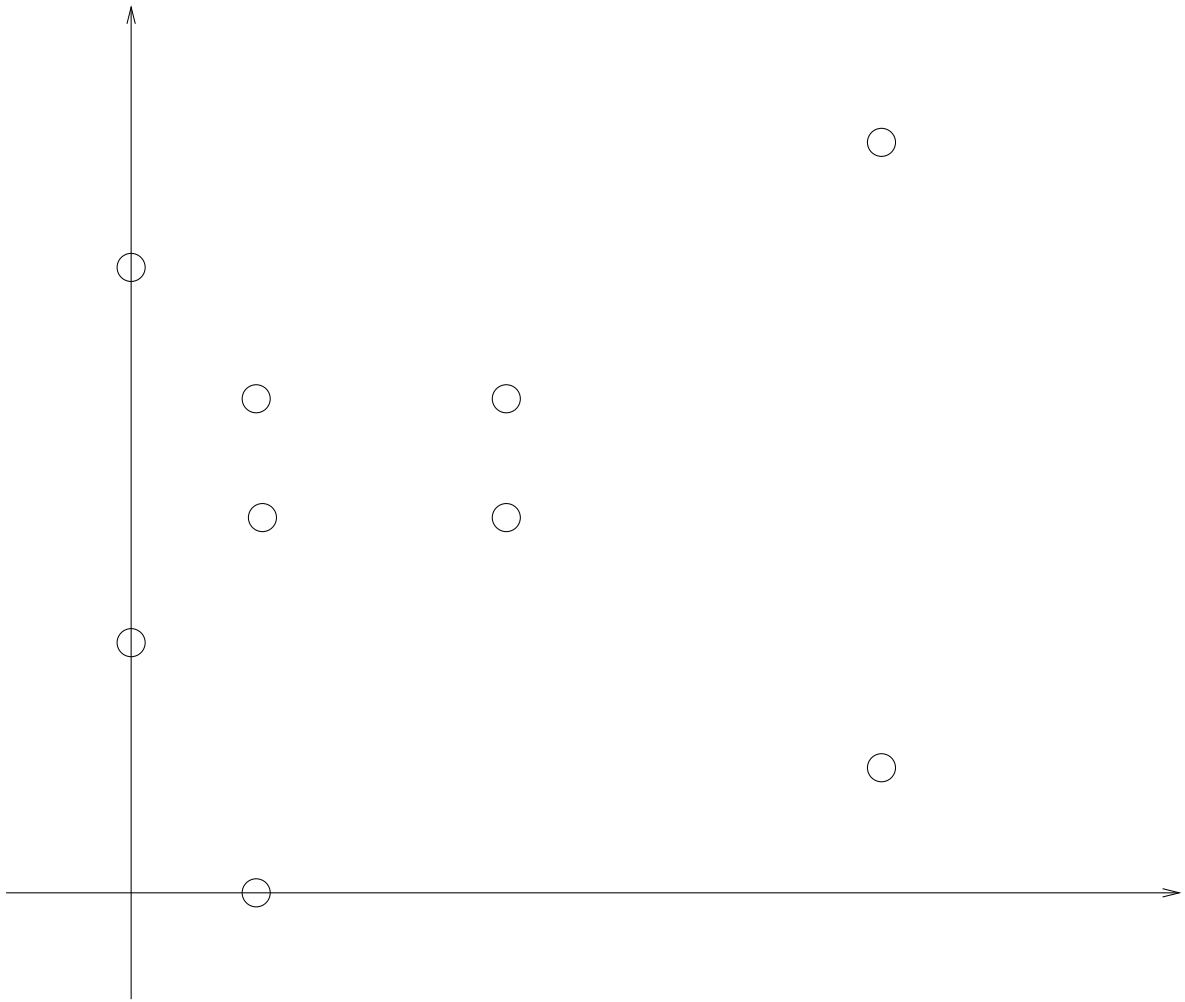
- Para uma determinda curva (a,b),...
- Para um determinado ponto
- Trabalhamos apenas com inteiros...
- ...e (mod) primo

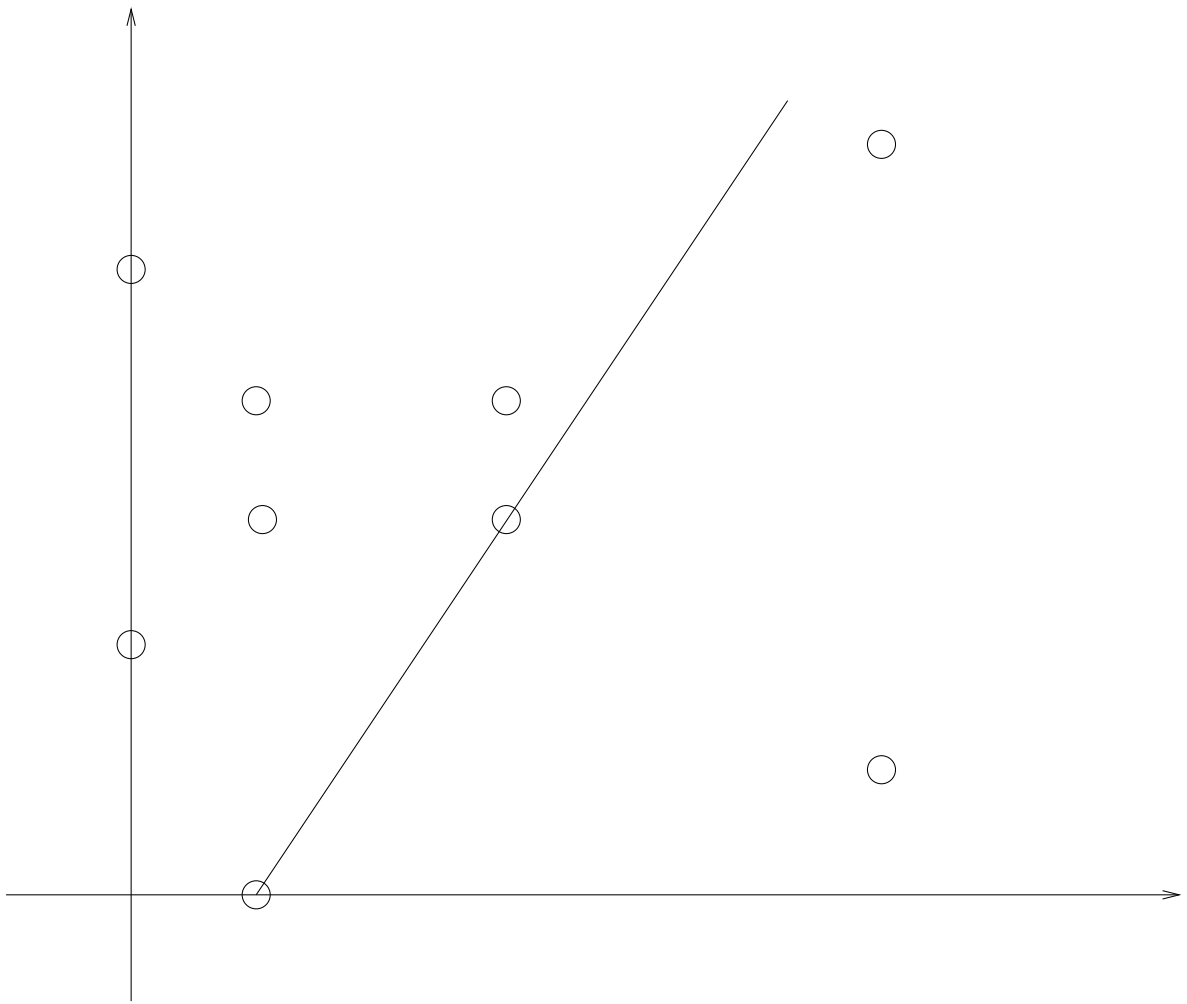
Exemplo

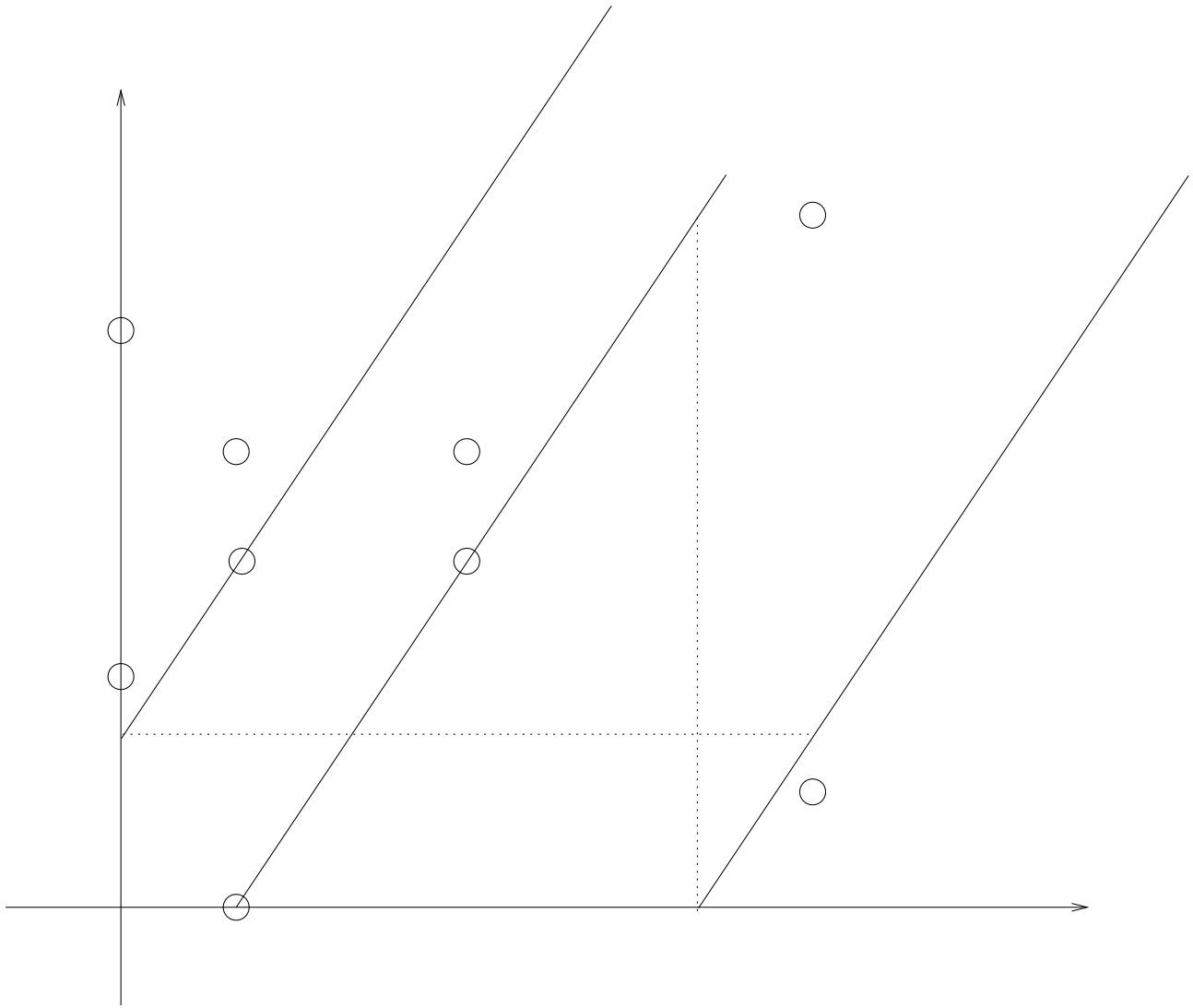
$$y^2 = x^3 + 2x + 4$$

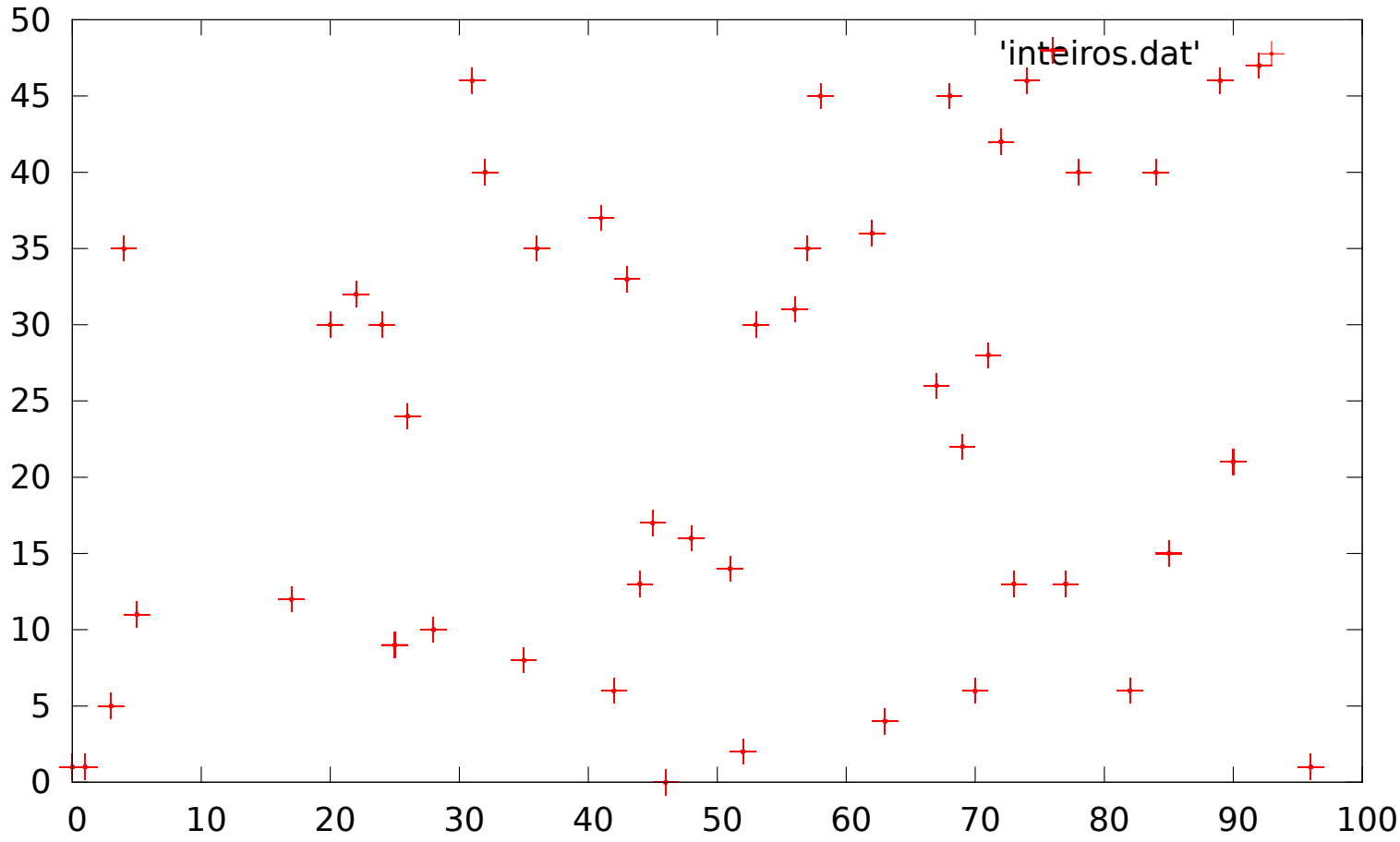
$$E(\mathcal{F}_7) = \{\infty, (0,2), (0,5), (1,0), (2,3), (2,4), (3,3), (3,4), (6,1), (6,6)\}$$

(exemplo de Guide to ECC - hankersn, Menezes e Vanstone)








'inteiros.dat'
50
45
40
35
30
25
20
15
10
5
0
0
10
20
30
40
50
60
70
80
90
100

- Estes pontos
- Junto com as operações definidas
- Formam um Corpo Finito
  Estrutura algébrica formada por: um conjunto finito, $+$ e $\times$ e a matemática é a mesma do 8o ano
- Fácil multiplicar por escalar....
- Difícil "dividir":
- Dado Q e P, determinar n tal que Q=nP...
- Já vimos isso em outro contexto...

- As "curvas" são públicas a, b e p
- Existe um ponto G, também público
- A gera k_A, $K_A = k_A G$
- B gera k_B, $K_B = k_B G$
- Cada um publica o seu K
- E o resto é DH!

- As “curvas” são públicas a, b e p
- Existe um ponto G, também público
- A gera k_A, $\mathcal{K}_A = k_A G$
- B gera k_B, $\mathcal{K}_\mathcal{B} = k_B G$
- Cada um publica o seu $\mathcal{K}$
- A: $S = k_A \mathcal{K}_\mathcal{B} = K_a k_B G$
- B: $S = k_B \mathcal{K}_\mathcal{A} = k_b k_a G$

SK

- Normalmente no ciframento A e B **compartilham** um segredo
- Chave secreta
- Esta chave secreta é usada para cifra e decifrar uma informação
- Vamos apresentar uma funcionalidade diferente

SK

- O protocolo define uma curva a, b, G e n
- A chave **privada** é um número aleatório (1 a n)
  Privada = sua, só sua
- $sk = random(1, n)$
- ou $sk = sha256('segredo')$

PK

- Alice:sk
- A chave pública é um "deslocamento/caminhada" nos pontos da curva
- $P_k = skG$
- P_k é um par ordenado, um ponto na curva a,b
- A P_k recebe este nome, justamente por ser pública
- (É basicamente o seu "endereço" Bitcoin, Ethereum,...)

Ciframento

- Quando Bob precisar mandar um segredo para Alice
- Nem sempre Bob a Alice tem uma oportunidade de ter um segredo compartilhado
- Bob conhece chave publica de Alice

Ciframento

- Alice:sk. Publico: $P_k = skG$
- Bob:
- $r = random$
- $R = rG$
- $S = rP_k = rskG$
- $k = \mathcal{H}(S)$
- A chave **simétrica/SECRETA** é o k
- O valor R é enviado junto com a mensagem cifrada....
- $R, \mathcal{E}_k(M)$

Deciframento

- Alice:sk, R
- Bob:$S = r.P_k = r.sk.G$ e $\mathcal{H}(S)$.
- Todos/Eva: $G, P_k(= sk.G), R(= r.G)$
- Alice: $S' = skR$
- $S' = sk.r.G$
- $S' = S$
- Alice obtem S, a chave simétrica....
- Eva?? So se souber a divisão....

IES

- Integrated Encryption Scheme
- Sistema Hibrido de ciframento
- ECC para enviar/definir uma chave secreta
- Cria um sistema assimétrico
- **FUNCIONALIDADE**
- Primeiros: DH e RSA

- Claus-Peter Schnorr/1989
- Protocolo Iterativo de Autenticação
- Conheço k (de um $K = kG$) sem revelar k
- $\mathcal{P}$ gera α aleatório e envia αG ao $\mathcal{V}$
- $\mathcal{V}$ gera um **desafio** c e envia para $\mathcal{P}$
- $\mathcal{P}$ responde $r = \alpha + c * k$
- $\mathcal{V}$ calcula $R = rG$ e $R' = \alpha G + cK$ e verifica se $R == R'$
- Tente provar **sem** conhecer o k
- (Tente provar em dois momentos diferente usando o mesmo α)

- Fita-Shamir
- Não iterativo
- $\mathcal{P}$
  1. Gera α aleatório e αG
  2. Gera o **desafio** $c = \mathcal{H}(\alpha G)$
  3. Faz $r = \alpha + c * k$
  4. Publica $(\alpha G, r)$
- $\mathcal{V}$:
  1. Gera o **desafio** $c' = \mathcal{H}(\alpha G)$
  2. Calcula $R = rG$ e $R' = \alpha G + c'K$ e verifica se $R == R'$

$$rG = (\alpha + cK)G$$

$$= (\alpha G) + (c * kG)$$

$$= \alpha G + cK$$

$$R = R'$$

- Alice gera k_A e $\mathbf{K_A} = \mathbf{k_A G}$
- $\mathbf{K_A}$ é a chave publica da Alice
  No BTC/Ethereum é o endereço
- Para assinar uma mensagem m
- Gera α aleatório e $\alpha\mathbf{G}$
- Gera o **desafio** $c = \mathcal{H}(m|\alpha\mathbf{G})$
- Faz $r = \alpha - c * k_A$
- Publica (c, r, m) ($\mathbf{K_A}$ já é publico!)

- Qualquer pessoa pode calcular
- $c' = \mathcal{H}(m|rG + c\mathbf{K_A})$
- E verificar se $c == c'$

$$rG = (\alpha - ck_A)G$$

$$= (\alpha G) - c\mathbf{K_A}$$

$$\alpha G = rG + c\mathbf{K_A}$$

Se

$$\mathcal{H}(m|\alpha G) == \mathcal{H}(m|rG + c\mathbf{K_A})$$

então

$$c = c'$$

!

- Complicado de entender a primeira vista....
- Mas são manipulações algébricas simples
- (e hash!)
- Este é um processo de assinatura e verificação de assinatura
- Somente A pode gerar (c, r, m)
- Que seja verificável por qquer outra pessoa
- Alice **assinou** m
- ECDSA