Criptografia Aplicada

Criptografia assimétrica - ECC

Sumário

- Definições básicas
- Criptografia em curvas elípticas (ECC)
- Eficiência e Segurança
- ECC na prática

Criptografia assimétrica

- Baseada no problema da fatoração de inteiros
- Baseada no problema do logarítmo discreto
- Baseada em curvas elípticas
    - conceitos matemáticos são mais difíceis de entender do que RSA ou Diffie-Hellman

Definições básicas

- **Curva elíptica**: conjunto de pontos que satisfaz a equação $\mathbf{y^2 = x^3 + ax + b}$
  - Chamamos esse conjunto de pontos (x,y) de E(a,b), para dados a e b.
  - Variáveis e coeficientes estão restritos a valores em um conjunto especial de elementos (ex: $\mathbb{Z}_p$)
  - Também definimos um ponto no infinito O
- **Exemplo**:
  - Curva E(-1,0) que satisfaz $y^2 = x^3 - x$ em $\mathbb{R}$
  - Curva E(-1, b) que satisfaz $y^2 = x^3 - x + b$ em $\mathbb{R}$, para b = 0, 1/10, 2/10, 3/10, 4/10, 5/10.

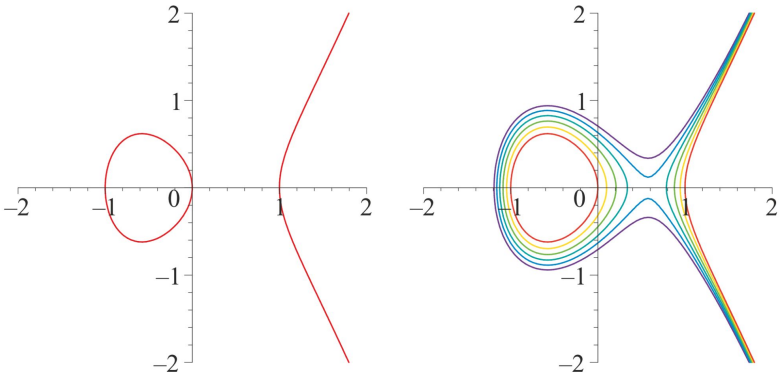


Imagem: E(-1,0) e E(-1, b).
Joachim von zur Gathen.
CryptoSchool. Cap 5

LabSEC
Laboratório de Segurança em Computação

cpqd

Definições básicas

- **Curva elíptica**: conjunto de pontos que satisfaz a equação $\mathbf{y^2 = x^3 + ax + b}$
  - Chamamos esse conjunto de pontos (x,y) de E(a,b), para dados a e b.
  - Variáveis e coeficientes estão restritos a valores em um conjunto especial de elementos (ex: $\mathbb{Z}_p$)
  - Também definimos um ponto no infinito O
- **Exemplo**:
  - Curva $E_{23}(1, 1)$ que satisfaz $y^2 = x^3 + x + 1$ em $\mathbb{Z}_{23}$.
    - $(0, 1) \in E_{23}(1, 1)$

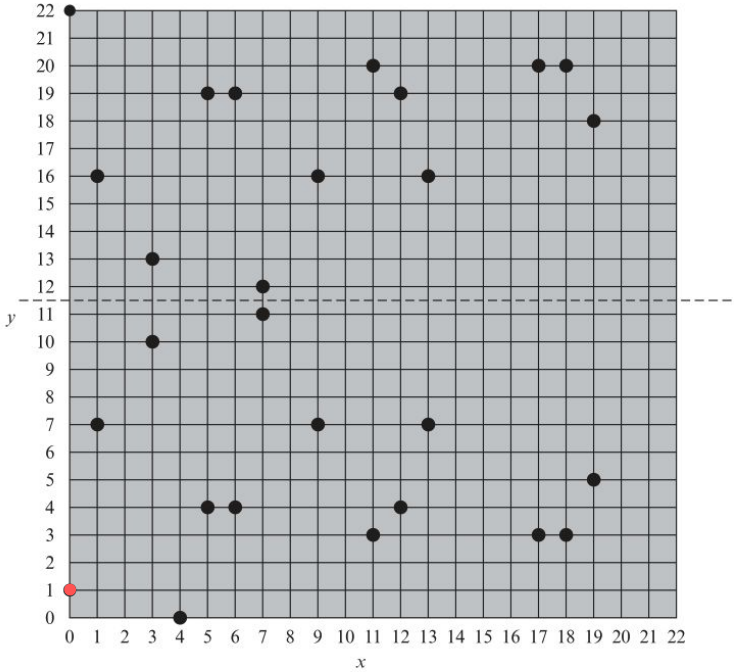


Figure 10.5 The Elliptic Curve $E_{23}(1, 1)$

Imagem: W. Stallings. *Cryptography and network security*. Cap 10.3

Definições básicas

- **Curva elíptica**: conjunto de pontos que satisfaz a equação $\mathbf{y^2 = x^3 + ax + b}$
  - Chamamos esse conjunto de pontos (x,y) de E(a,b), para dados a e b.
  - Variáveis e coeficientes estão restritos a valores em um conjunto especial de elementos (ex: $\mathbb{Z}_p$)
  - Também definimos um ponto no infinito O
- **Exemplo**:
  - Curva $E_{23}(1, 1)$ que satisfaz $y^2 = x^3 + x + 1$ em $\mathbb{Z}_{23}$.

Table 10.1 Points (other than *O*) on the Elliptic Curve $E_{23}(1, 1)$

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

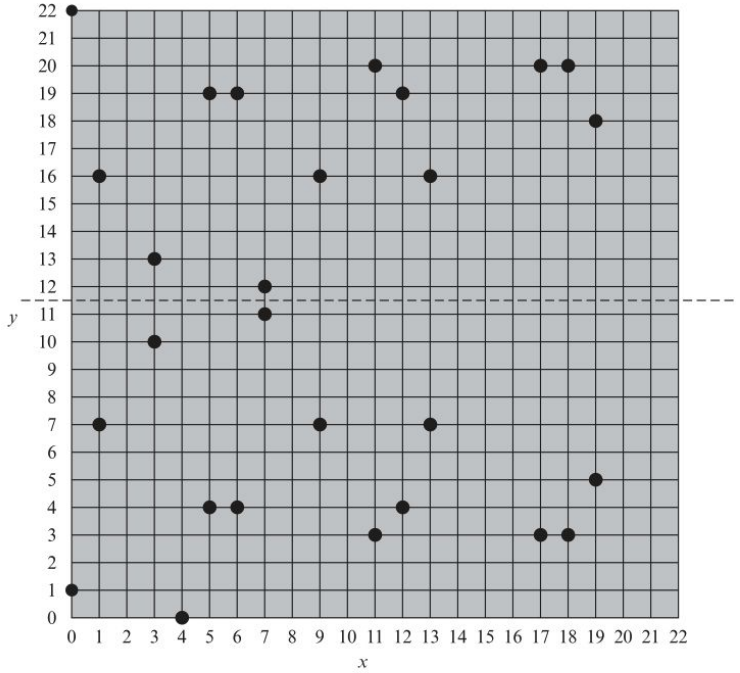


Figure 10.5 The Elliptic Curve $E_{23}(1, 1)$

Imagens: W. Stallings. *Cryptography and network security*. Cap 10.3

Propriedades

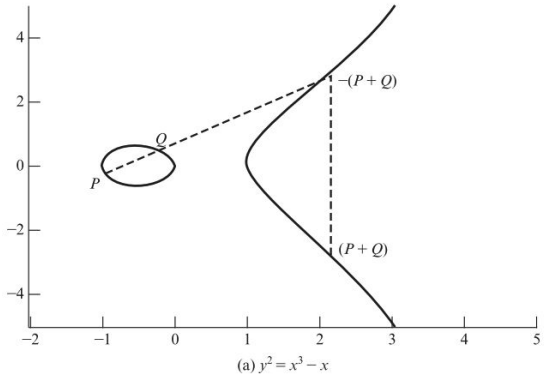


Figure 10.4 Example of Elliptic Curves

- E(a,b): conjunto de pontos que satisfaz a equação $\mathbf{y^2 = x^3 + ax + b}$
- **Soma** de dois pontos P e Q dão um outro ponto R na curva (P+Q = R).
- **Multiplicação** pode ser definida como somas consecutivas: P+P = 2P e o resultado está na curva.
- P + O = P (identidade)
- Se P = (x,y), então -P = (x, -y) (inversa)
- P + (-P) = O

Imagem: W. Stallings. *Cryptography and network security*. Cap 10.3

Sumário

- Definições básicas
- **Criptografia em curvas elípticas (ECC)**
- Eficiência e Segurança
- ECC na prática

Criptografia em Curvas Elípticas

- Resposta ao tamanho de chaves RSA
  - que tem crescido nos últimos anos
- Mesma segurança com chaves menores
- Computações mais eficientes
- Baseada em funções matemáticas simples de calcular mas difíceis de reverter

Table 9.3 Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie–Hellman	No	No	Yes
DSS	No	Yes	No

Criptografia em Curvas Elípticas

- A adição em ECC é a contrapartida da multiplicação modular no RSA.
- A multiplicação em ECC é a contrapartida da exponenciação modular.
- Precisamos de um **problema difícil** em ECC, assim como temos a fatoração ou o problema do logaritmo discreto.

Criptografia em Curvas Elípticas

- No protocolo Diffie-Hellman fazemos $Y = m^k \bmod p$
  - multiplicamos m por ele mesmo k vezes
  - um atacante precisa determinar k (log discreto)
- Em ECC, fazemos somas sucessivas
  - Calculamos $Q = kP$, onde Q e P são pontos em $E_p(a,b)$ (ou seja, em $\mathbb{Z}_p$) e $k < p$
  - é fácil calcular Q dado k e P
  - é difícil determinar k dado Q e P
    - Esse é o **problema do logaritmo discreto em curvas elípticas**
    - k é muito grande, tornando força-bruta inviável

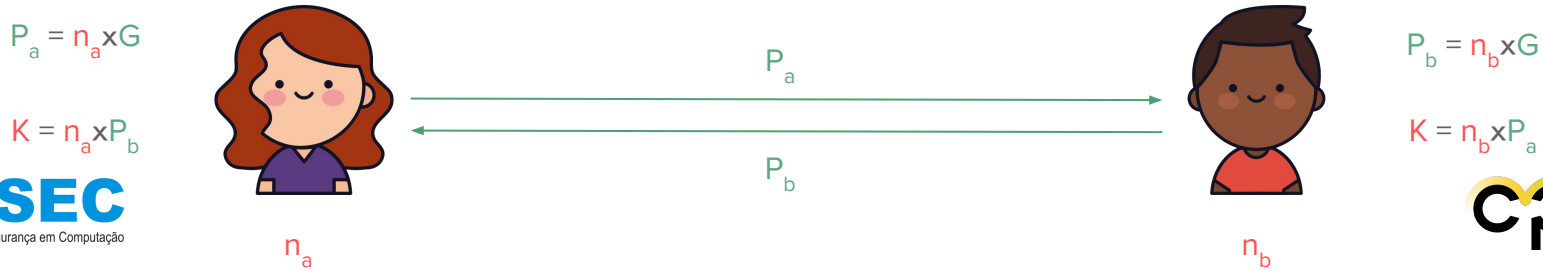
Troca de chaves em ECC

Parâmetros

- **Parâmetros públicos:**
  - q número primo ou na forma 2^m
  - parâmetros a e b para a curva $E_q(a,b)$
  - ponto G = (x_1, y_1) em $E_q(a,b)$ cuja ordem é um número grande n
    - ordem: menor inteiro n tal que nG = O
- **Parâmetros privados**:
  - n_a e n_b números aleatórios < n

Algoritmo:

- **Cálculo dos valores públicos:**
  - Alice calcula: $P_a = n_a x G$
  - Bob calcula: $P_b = n_b x G$
- **Cálculo do segredo:**
  - Alice calcula: $K = n_a x P_b$
  - Bob calcula: $K = n_b x P_a$



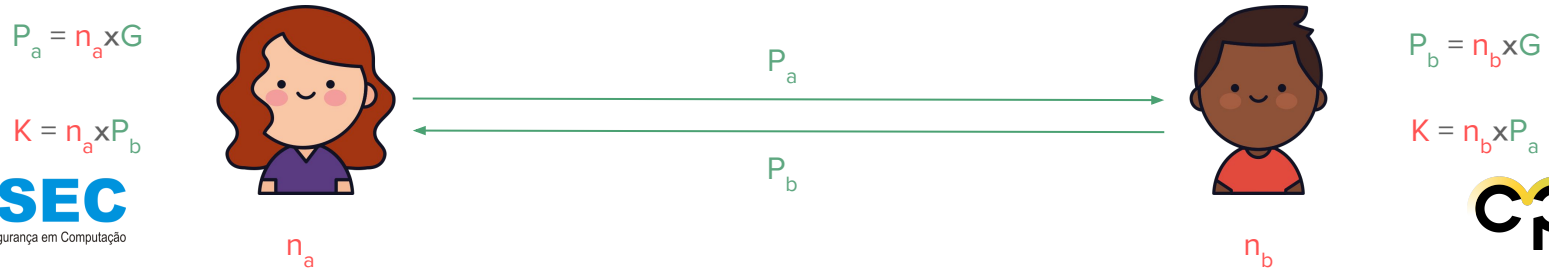
LabSEC
Laboratório de Segurança em Computação

cpqd

Troca de chaves em ECC

- **Corretude:** $K = n_a x P_b = n_a x (n_b x G) = n_b x (n_a x G) = n_b x P_a$
  - portanto, ambos calculam a mesma chave K

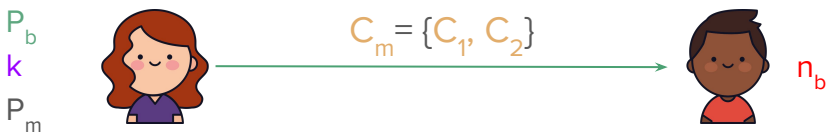
- **Segurança**: um atacante deveria ser capaz de calcular n, dado G e nG.
  - com esse valor secreto, ele é capaz de calcular K
  - encontrar n significa resolver o problema do logaritmo discreto em curvas elípticas



Cifragem e decifragem

Geração de chaves:

- Considere um ponto $G = (x_1, y_1)$ em $E_q(a,b)$
- Bob calcula:
  - chave privada n_b
  - chave publica $P_b = n_b x G$



$C_m = \{kG, P_m + kP_b\}$

$P_m = C_2 - n_b C_1$

Cifragem:

- Alice escolhe um valor secreto k e codifica a mensagem m em um ponto P_m
- Alice cifra a mensagem usando P_b
- $C_m = \{kG, P_m + kP_b\} = \{C_1, C_2\}$

Decifragem:

- Bob **decifra** a mensagem usando a sua chave secreta n_b
- $P_m = C_2 - n_b C_1$

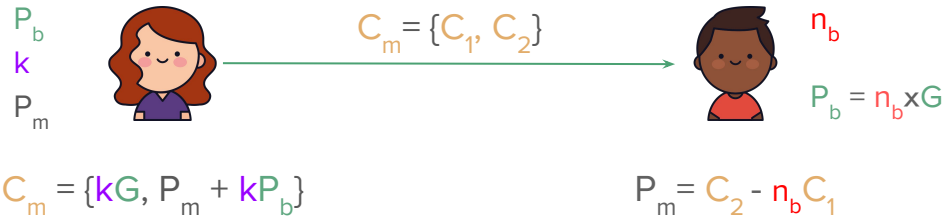
LabSEC
Laboratório de Segurança em Computação

cpqd

Cifragem e decifragem

- **Corretude:** $C_2 - n_b C_1 = (P_m + kP_b) - n_b(kG) = P_m + k(n_b G) - n_b(kG) = P_m$
  - ao utilizar a chave secreta n_b, é possível recuperar a mensagem original P_m

- **Segurança**: Alice mascarou a mensagem P_m ao adicionar kP_b a ela.
  - para remover kP_b e obter P_m, um atacante precisaria descobrir k dado kP_b
  - descobrir k significa resolver o problema do logaritmo discreto em curvas elípticas



Sumário

- Definições básicas
- Criptografia em curvas elípticas (ECC)
- **Eficiência e Segurança**
- ECC na prática

Segurança e Eficiência

- A segurança de ECC depende da dificuldade de determinar k dado kP e P
  - Elliptic Curve Discrete Logarithm Problem (ECDLP)
- O ataque mais rápido é conhecido como método *Pollard rho*
- Recomendação do **NIST** é a de utilizar chaves entre 256 e 512 bits para ECC
- ECC requer chaves muito menores para o mesmo nível de segurança do RSA
- Por usar chaves menores, requer menos poder computacional
  - ideal para dispositivos limitados

Comparação do tamanho de chaves

Table 10.3 Comparable Key Sizes in Terms of Computational Effort for Cryptanalysis (NIST SP-800-57)

Symmetric Key Algorithms	Diffie–Hellman, Digital Signature Algorithm	RSA (size of n in bits)	ECC (modulus size in bits)
80	$L = 1024$ $N = 160$	1024	160–223
112	$L = 2048$ $N = 224$	2048	224–255
128	$L = 3072$ $N = 256$	3072	256–383
192	$L = 7680$ $N = 384$	7680	384–511
256	$L = 15{,}360$ $N = 512$	15,360	512+

Note: L = size of public key, N = size of private key.

Imagem: W. Stallings. *Cryptography and network security*. Cap 10.4

NIST SP 800-57

Recomendações de uso

- O SP 800-57 recomenda que, a partir de 2031, o RSA seja usado com chaves a partir dos 3072 bits e ECC com chaves entre 256 e 512 bits.

Table 4: Security strength time frames

Security Strength		Through 2030	2031 and Beyond
< 112	Applying protection	Disallowed	
	Processing	Legacy use	
112	Applying protection	Acceptable	Disallowed
	Processing		Legacy use
128	Applying protection and processing information that is already protected	Acceptable	Acceptable
192		Acceptable	Acceptable
256		Acceptable	Acceptable

Sumário

- Definições básicas
- Criptografia em curvas elípticas (ECC)
- Eficiência e Segurança
- **ECC na prática**

Aplicações

- Utilizadas em ambientes com capacidades limitadas de comunicação
- Uso na prática
  - TLS
  - Bitcoin
  - iMessage

Curvas do NIST

- Quais curvas elípticas devemos utilizar?
- O SP 800-186 traz a medida de segurança e recomendação de curvas seguras
- As curvas são identificadas por um nome, como P-256, Edwards448, Curve448 e suas especificações são dadas no documento
- Através destes nomes, conseguimos identificá-las nas bibliotecas criptográficas

Atividade: Gerando chaves com o openssl

- Verifique as curvas disponíveis no openssl:
  ```
  openssl ecparam -list_curves
  ```
- Gere a chave privada com a curva escolhida:
  ```
  openssl ecparam -genkey -name <nome_da_curva> -out <nome_chave_privada>.pem
  ```
- Extraia a chave pública:
  ```
  openssl ec -in <nome_chave_privada>.pem -pubout -out <nome_chave_pública>.pem
  ```
- Visualize as chaves:
  ```
  openssl ec -in <nome_chave_privada>.pem -text -noout
  openssl ec -in <nome_chave_pública>.pem -text -noout -pubin
  ```
- Compare as chaves geradas com as chaves do RSA
  - Obs: se escolher secp224r1 terá uma chave com o mesmo nível de segurança do RSA 2048 que geramos na aula passada. Compare os tamanhos das chaves.

Referências

- W. Stallings. *Cryptography and network security*. 7a edição.
  - Curvas Elípticas: 10.4
- Joachim von zur Gathen. *CryptoSchool*. 1a edição.
  - Curvas Elípticas: 5
- imagem: Flaticon.com
- SP 800-57
- SP 800-186