Coordenadoria do Curso Ciência da Computação
Universidade Estadual do Mato Grosso do Sul

UM ESTUDO DA CRIPTOGRAFIA DE CURVAS ELÍPTICAS

Geisiane Martini Ferreira e Tiago Antonio Pereira Borges

Adriana Betânia de Paula Molgora (Orientadora)

Dezembro

2013

UM ESTUDO DA CRIPTOGRAFIA DE CURVAS ELÍPTICAS

Geisiane Martini Ferreira e Tiago Antonio Pereira Borges

Monografia da disciplina Projeto Final de Curso, Coordenadoria do Curso Ciência da Computação, Universidade Estadual do Mato Grosso do Sul, como requisito para obtenção do título de Bacharel em Ciência da Computação.

Adriana Betânia de Paula Molgora

(Orientadora)

Coordenadoria do Curso de Ciência da Computação

Universidade Estadual de Mato Grosso do Sul

Um Estudo da Criptografia de Curvas Elípticas

Geisiane Martini Ferreira e Tiago Antonio Pereira Borges

Novembro 2013

Banca Examinadora:

Adriana Betânia de Paula Molgora (Orientadora)

Ciência da Computação

Fabrício

Ciência da Computação

Odival Faccenda

Ciência da Computação

Resumo

Pode-se definir criptografia como as formas de codificar uma mensagem onde somente o destinatário tem ferramentas suficientes para interpretá-la. Existem diversos métodos de criptografia e, dentre esses, figura a criptografia baseada em Curvas elípticas que oferece uma forma segura para criptografar e descriptografar mensagens. Este trabalho apresenta um estudo teórico e prático sobre esta criptografia.

Palavra-Chave: *Criptografia Curvas elípticas.*

Abstract

You can set encryption as ways to encode a message in which only the recipient has enough tools to interpret it. There are several encryption methods, and among these, figure based on Elliptic curves cryptography that offers a safe way to encrypt and decrypt messages. This paper presents a theoretical and practical study on this encryption.

Keyword: *Elliptic Curves Cryptography*.

Agradecimentos

Agradeço primeiramente a Deus, pois, sem ele jamais chegaria até aqui. Aos meus pais, Manoel e Ana Paula, e meu namorado Rogers que me proporcionaram apoio durante minha graduação.

Aos meus amigos Aline, Priscila, Tiago, Juliana, Adriana, Natalia e Gleyciane que sempre me ajudaram em tudo durante a graduação e que me incentivaram a continuar.

À Professora Adriana Betânia de Paula Molgora pela orientação, incentivo dado para o término desse trabalho.

E a todos os demais colegas e Professores que contríbuiram para o sucesso desse trabalho.

Geisiane Martini Ferreira.

Primeiramente agradeço a Deus, sem ele jamais conseguiria chegar até aqui.

Agradeço a minha mãe, Maria, por ter me apoiado em minhas decisões e se dedicado a minha formação acadêmica.

A minhas irmãs, Ana e Vera, e minha tia, Madalena, por me apoiarem e me ajudarem em todos os momentos da minha vida.

Aos meus amigos Aline, Priscila, Geisiane, Juliana, Adriana, Natalia, Simone e Gleyciane por terem me ajudado ao longo da minha formação acadêmica.

A professora Adriana Betânia de Paula Molgora, por ter acreditado que seriamos capazes de realizar este trabalho, ter paciência conosco e se dedicar em nos orientar a qualquer momento.

Agradeço a todos os professores que tive ao longo de minha vida.

Tiago Antonio Pereira Borges.

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Algoritmos

Capítulo 1

Introdução

Desde a antiguidade, existe a preocupação de proteger o processo de troca de mensagens entre emissor e destinatário. De acordo com Quaresma e Lopes [14] entre os séculos 100-44 a.C., o Imperador Romano Júlio César desenvolveu algo para se comunicar com os seus Generais que era uma cifra simples. A mensagem era a seguinte:

> Na mensagem original cada letra é deslocada três posições para a direita, considerando-se que o alfabeto se fecha sobre si próprio, isto é, que após a última letra vem a primeira; o receptor da mensagem só tem que deslocar cada letra três posições para a esquerda para obter a mensagem original.

A criptografia surgiu com o intuito de proporcionar a segurança na troca de informações/mensagens, sendo um mecanismo para a proteção desses dados. Existem diversos tipos de criptografia e, dentre esses figura a Criptografia baseada em curvas elípticas.

A Criptografia baseada em curvas elípticas foi desenvolvida por Neal Koblitz e Victor Miller em 1985 Sangalli [13]. O fato de a criptografia baseada em curvas elípticas proporcionar o mesmo nível de segurança que os outros algoritmos, mas com chaves menores, é o que tornou interessante o estudo dessa criptografia Sangalli [13].

A proposta desse trabalho é apresentar um estudo teórico e prático do funcionamento do sistema criptográfico baseado em curvas elípticas.

1.1 Objetivo

Este trabalho tem como objetivo geral realizar um estudo da criptografia de curvas elípticas a fim de disponibilizar conhecimentos sobre a mesma.

1.1.1 Objetivos Específicos

Os Objetivos específicos são:

- Estudar os conceitos matemáticos necessários para o entendimento do algoritmo de criptografia baseada em Curvas Elípticas.
- Estudar os algoritmos que realizarão a codificação e decodificação da criptografia baseada em Curvas Elípticas.

1.2 Justificativa e Motivação

A criptografia baseada em curvas elípticas apresenta uma grande segurança dos dados/mensagens, pois de acordo com a complexidade do seu algoritmo, o processo de decifragem é praticamente intratável se não houver o conhecimento da chave secreta. Para entender como é o funcionamento desse sistema é necessário adquirir conhecimentos matemáticos e computacionais sobre o mesmo.

Este trabalho busca aprimorar o entendimento do sistema criptográfico baseado em curvas elípticas, a fim de disponibilizar conhecimentos que poderão servir como base para trabalhos futuros.

1.3 Metodologia

Para o alcance dos objetivos, o trabalho foi distribuído nas seguintes etapas:

1. Estudo dos conceitos matemáticos fundamentais e de curvas elípticas, através de revisão bibliográfica.
2. Implementação dos algoritmos de codificação e decodificação da criptografia de curvas elípticas.
3. Documentação dos estudos realizados.

1.4 Organização do Texto

Capítulo 2:

Fundamentação Teórica

No capítulo 2, são apresentados os conceitos matemáticos básicos utilizados no processo de criptografia das curvas elípticas.

Capítulo 3:

Curvas Elípticas

No capítulo 3, são relatados os conceitos teóricos das curvas elípticas necessários para o melhor entendimento dessa criptografia.

Capítulo 4:

Criptografia

No capítulo 4 é descrito o funcionamento do método criptográfico de curvas elípticas, bem como alguns algoritmos envolvidos nesse processo.

Capítulo 5:

Implementação

No capítulo 5 é apresentada, a implementação dos algoritmos de codificação, decodificação e de inversor, explicando detalhadamente seu funcionamento.

Capítulo 6:

Testes

No capítulo 6 são apresentados os testes realizados e os resultados obtidos.

Capítulo 2
Fundamentação Teórica

Para uma melhor compreensão da Criptografia baseada em Curvas Elípticas, faz-se necessário o estudo de alguns conceitos básicos da Teoria dos Números. Esses conceitos podem ser encontrados em [1,2, 3, 8].

2.1 Números Primos

Definição 2.1. *Um número natural $p > 1$ é denominado número primo se seus únicos divisores são p e 1.*
Um número que não satisfaz a condição de primalidade diz-se composto.
Exemplo 2.1. O número 11 é primo, pois só é divisível por 1 e 11.
Exemplo 2.2. O número 20 é divisível por 1,2,4,5,10 e 20. Logo, 20 é um número composto.

2.2 Unicidade da Fatoração

Todo número inteiro $n \geq 2$, pode ser escrito como um produto de números primos. Essa decomposição é única, a menos da ordem dos fatores.

Exemplo 2.3. $30 = 2 \cdot 3 \cdot 5$, isto é, 30 é a multiplicação dos números primos 2, 3 e 5.

2.3 Divisibilidade

Definição 2.2. *Sejam a e b pertencentes ao conjunto dos inteiros. Diz-se que a divide b, se existe um inteiro c tal que $b = a \cdot c$. Quando a divide b, diz-se também que a é divisor de b, ou a é fator de b ou b é múltiplo de a. Se a divide b, então escreve-se por $a \mid b$.*

Exemplo 2.4. 4 divide 8, pois $4 \cdot 2 = 8$

Proposição 2.1. Propriedade da divisibilidade. *Para todos a, b, c e d em $\mathbb{Z}$, tem-se as seguintes propriedades:*

i) $a \mid 0$ e $a \mid a$

ii) Se $a \mid b$ e $b \mid c$, então $a \mid c$

iii) Se $a \mid b$ e $b \mid c$, então $a \cdot c \mid b \cdot d$

iv) Se $a \mid (b + c)$ e $a \mid b$, então $a \mid c$

v) Se $a \mid b$ e $a \mid c$, então $a \mid (b \cdot x + c \cdot y)$ $\forall$ $x,y \in \mathbb{Z}$

vi) Se $a \mid b$ e $b \mid a$, então $a = \pm b$.

2.3.1 MDC – Máximo Divisor Comum

Definição 2.3. *O Máximo Divisor Comum de dois inteiros a e b (a e b devem ser não nulos), é um inteiro positivo d, escreve-se* $d = mdc\ (a, b)$ *que satisfaça as seguintes condições:*

i) *d é um divisor comum de a e b.*

ii) *Se c | a e c | b, então c | d*

Exemplo 2.4. *mdc* (36, 90) é 18, *mdc* (6,12) é 6, *mdc* (12, 20) é 4.

2.4 Inteiros Módulo n

Definição 2.4. *Seja n um inteiro positivo. Tem-se que dois inteiros positivos a e b são congruentes módulo n, se a e b deixam o mesmo resto quando divididos por n. Se a e b são congruentes módulo n, denota-se* $a \equiv b \pmod{n}$.

Exemplo 2.6. $42 \equiv 31 \pmod{11}$, pois 11 divide 42 – 31 = 11, ou ainda porque na divisão de 42 por 11 obtém-se o mesmo resto 9

Exemplo 2.7. $54 \equiv 3 \pmod{17}$, pois 17 divide 54 – 3 = 51, ou ainda porque na divisão de 54 por 17 tem-se o mesmo resto 3.

Proposição 2.1. *Tem-se que* $a \equiv b \pmod{n}$*se, e somente se,* $n \mid (a - b)$*. Isto é se* $(a - b) = nk$ *para algum inteiro k.*

Propriedades da Congruência: Para todos *a, b, c, d, m* e *n* em $\mathbb{Z}$, com $m \geq 1$ e $n > 1$, tem-se as seguintes propriedades:

i) *Se* $a \equiv b \pmod{n}$.

ii) Se $a \equiv b \pmod{n}$então $b \equiv a \pmod{n}$.

iii) Se$a \equiv b \pmod{n}$, então $a \equiv c \pmod{n}$.

iv) Se $a \equiv b \pmod{n}$e$c \equiv d \pmod{n}$, então $a + c \equiv b + d \pmod{n}$.

v) Se $a \equiv b \pmod{n}$e$c \equiv d \pmod{n}$,então $ac \equiv bd \pmod{n}$

vi) Se$a \equiv b \pmod{n}$, então $a^m \equiv b^m \pmod{n}$

2.5 Grupos

Definição 2.5. *Seja G um conjunto não vazio onde está determinada uma operação entre pares de G, denotado por:*

$$* : G \times G \to G$$
$$(x, y)\ \alpha\ \ x * y$$

Tem-se que o par $(G, *)$ é um grupo, se as propriedades a seguir são válidas:

i) **Associativa:** Quaisquer que sejam *a,b* e *c* em *G* diz-se que $(a * b) * c = a * (b * c)$.

ii) **Existência de elemento neutro:** Existe um elemento e em *G* tal que $a * \mathrm{e} = e * a = a$.

iii) **Existência de elemento simétrico:** Para cada $a \in G$, existe $b \in G$ tal que $a * b = b * a = e$.

Vale ainda em um grupo *(G, ∗)* verificar a propriedade:

iv) **Comutativa:** $a * b = b * a$, para qualquer que seja *a, b* $\in G$, tem-se que o grupo *(G, ∗)* é um grupo abeliano ou comutativo.

Exemplo 2.8. Considerando o conjunto dos números reais definido pela operação $*$ e por $x * y = x + y - 3$. Mostre que $(\mathbb{R}, *)$ é um grupo comutativo.

i) **Associatividade:** *(x ∗ y) ∗ z = (x + y – 3) ∗ z*

$\Rightarrow (x + y - 3) + z - 3 \Rightarrow x + (y + z - 3) - 3 \Rightarrow x * (y + z - 3)$

$(x * y) = x * (y * z)$.

ii) **Existência de elemento neutro:** $x * e = e * x = x$

$x * e = x \Rightarrow x + e - 3 = x \Rightarrow e = 3$.

iii) **Existência de elemento simétrico:** $x * x' = e = x' * x$

$x * x' = 3 \Rightarrow x + x' - 3 = 3 \Rightarrow x' = 6 - x.$

iv) **Comutatividade:** $x * y = x + y - 3 \Rightarrow y + x - 3 = y * x$

Portanto ($\mathbb{R}$, $*$) é um grupo abeliano.

2.6 Subgrupos

Definição 2.6. *Seja (G, $*$) um grupo. Existe um subgrupo H, não vazio, de G é um subgrupo de G, logo que:*

i) $\forall\ h_1, h_2 \in H \Rightarrow h_1 * h_2 \in H$

ii) *(H, $*$)* também é grupo, isto é, dados h_1, h_2, $h_3 \in H$, as propriedades a seguir são satisfeitas:

1) $h_1 * (h_2 * h_3) = (h_1 * h_2) * h_3$.

2) $\exists\ e_H \in H,$, tal que $e_H * h_1 = h_1 * e_H = h_1$.

3) Para cada $h \in H$, existe $K \in H$ tal que $h * K = K * h = e_H$.

Exemplo 2.9. O conjunto $\mathbb{Z}$ é um subgrupo de o grupo abeliano ($\mathbb{Q}$,+), pois

1) Dados a,b $\in \mathbb{Z} \Rightarrow$ a + b $\in \mathbb{Z}$
2) ($\mathbb{Z}$,+) é um grupo, sendo $e = 0$ e $x' = -x$.

Teorema 2.1 (Teorema de Lagrange)

Diz-se *G* um grupo finito e *H* um subgrupo de *G*, então |*H*| é divisor de |*G*|, isto é, a ordem de *H* é um divisor da ordem de *G*.

2.7 Anéis

Definição 2.7. *Anel é um conjunto denotado como A, em que os elementos podem ser adicionados e multiplicados (ou seja, dadas duas operações (x,y) → x + y e (x,y) → xy aos pares de elementos de A em A), satisfazendo as seguintes condições:*

i) Para todo *x* e *y* $\in A$ tem-se a comutatividade da soma:

$$x + y = y + x$$

ii) Para todos *x* e *y* $\in A$ tem-se a associatividade da soma:

$$(x + y) + z = x + (y + z)$$

iii) Existência do elemento e em A tal *que* $x + e = x$ para todo $x \in A$. Quando $e = 0$ é chamado elemento neutro da adição.

iv) Para todo elemento $x \in A$ existe um elemento y em A tal que $x + y = 0$. Quando ocorre de $y = -x$ ele também é dito como simétrico de x.

v) Para todo $x, y, z \in A$ tem-se a associatividade da multiplicação:

$$(xy)z = x(yz)$$

vi) Para todo $x, y, z \in A$ tem-se a distributividade da multiplicação à direita e esquerda:

$$x(y + z) = xy + x.z$$

$$(y + z)x = yx + z.x$$

Observação 2.1. *A multiplicação não necessita ser comutativa. Quando isto ocorre, diz-se que A é um anel comutativo.*

Observação 2.2. *Na multiplicação um anel não requer um elemento neutro, ou seja, um elemento y tal que xy = yx = x para todo x ∈ A. Este elemento é conhecido de unidade do anel e denotado por 1. Quando acontece de um anel A ter um elemento neutro da multiplicação diz-se que A é um anel com unidade.*

Observação 2.3. *Os elementos não nulos de um anel não precisam ter inversos multiplicativos, isto é, y é inverso multiplicativo de x se, e somente se, xy = yx = 1. Um anel é chamado invertível de A ou unidade de A, quando os elementos de um anel apresenta inverso multiplicativo. Usa-se a notação U (A) = { x ∈A| x é uma unidade de A}.*

Exemplo 2.10. Considerando as operações $*$ e Δ em $\mathbb{Q}$ definidas por*: $x * y = x + y - 3$ e $x \Delta y = x + y - \frac{xy}{3}$*, mostre que $(\mathbb{Q}, *, \Delta)$ é um anel.

Como visto no Exemplo 2.9, $x * y = x + y - 3$ é um grupo abeliano. Para provar que é anel basta mostrar que vale as propriedades (v) e (vi).

v) Associatividade, Δ: $(x \Delta z) = \left(x + y - \frac{xy}{3}\right) \Delta z$

$$\Rightarrow x + y - \frac{1}{3}xy + z - \frac{1}{3}\left(x + y - \frac{1}{3}xy\right).z$$

$$\Rightarrow x + \left(y + z - \frac{1}{3}yz\right) - \frac{1}{3}x\left(y + z - \frac{1}{3}yz\right) = x \Delta (y \Delta z)$$

vi) Δ é distributiva em relação a $*$: $(x * y) \Delta z = x \Delta (x + y - 3) \Delta z$

$$\Rightarrow x + y - 3 + z - \frac{1}{3}(x + y - 3)z$$

$$\Rightarrow \left(x + z - \frac{1}{3}xy\right) + \left(y + z - \frac{1}{3}yz\right) - 3$$

$$\Rightarrow (x \Delta z) + (y \Delta z) - 3 = (x \Delta z) * (y \Delta z).$$

Logo, é anel.

2.8 Corpos

Definição 2.8. *Um anel K, comutativo com unidade, é chamado de corpo, se todo o elemento não nulo de K possuir simétrico com relação à multiplicação. Ou seja,*

$$\forall\, a \in K \text{ tal que } a \neq 0, \text{ então } \exists\, b \in K \text{ tal que } a \cdot b = 1.$$

Exemplo 2.11 $(\mathbb{Z}_5, +, .)$ é um corpo?

Sabendo que $\mathbb{Z}_5 = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}\}$, deve-se verificar se $\overline{1}, \overline{2}, \overline{3}$ e $\overline{4}$ tem simétrico multiplicativo.

Para isso faz-se:

mdc(5, 1) = 1

mdc(5, 2) = 1

mdc(5, 3) = 1

mdc(5, 4) = 1

Logo, todos os elementos não nulos de $\mathbb{Z}_5$ possuem inverso, isto é, $\mathbb{Z}_5$ é um corpo.

2.9 Teoremas de Fermat

Lema 2.1. Se *p é um número primo e a, b* $\in \mathbb{Z}$ *, então* $(a+b)^n \equiv a^n + b^n \pmod{p}$.

Teorema 2.2 (Teorema de Fermat)

Seja p um número primo e a um número inteiro, então $a^p \equiv a \pmod{p}$..

Teorema 2.3 (Pequeno Teorema de Fermat)

Seja p um número primo que não divide o inteiro a, então $a^{p-1} \equiv 1 \pmod{p}$.

Exemplo 2.12. $2^{16} \equiv 1 \pmod{17}$ e $7^{10} \equiv 1 \pmod{11}$.

2.10 Problema do logaritmo discreto (PLD)

Definição 2.9. *Seja um grupo G e y, $\alpha \in G$ tal que y é potência de α. Dizemos que o logaritmo discreto de y na base α é o menor inteiro não negativo x tal que $\alpha^x = y$, denotado por $\log_\alpha y = x$.*

O problema do logaritmo discreto assegura que não é possível definir *x* em um tempo computacionalmente razoável.

Capítulo 3
Curvas Elípticas

Este capítulo aborda os conceitos básicos das curvas elípticas, suas propriedades e a álgebra que envolve essas curvas. Os conceitos aqui abordados são baseados em [4, 5, 6, 7, 8].

3.1 Definição

É importante frisar que curvas elípticas não são elipses. Elas recebem esse nome, porque surgiram do estudo do comprimento de arco de uma elipse.

Podem ser definidas sobre um corpo *K*, como exemplo, o corpo dos números complexos. As curvas são descritas por uma equação cúbica do tipo

$$ax^3 + bx^2y + cxy^2 + dy^3 + ex^2 + fxy + gy^2 + hx + iy + j = 0 \qquad (3.1)$$

onde *a,b, ..., j* são elementos de um corpo *k*. Fazendo-se uma mudança apropriada de variáveis, uma curva elíptica geral pode ser escrita na forma normal de Weierstrass, sobre um corpo de característica diferente de 2 e 3, por:

$$y^2 = f(x) = x^3 + ax + b \qquad (3.2)$$

onde

$$4a^3 + 27b^2 \neq 0 \qquad (3.3)$$

de modo que *f* não tenha raízes múltiplas.

Existem curvas que assumem diferentes formas, dependendo dos parâmetros utilizados. A seguir são apresentadas figuras que ilustram alguns exemplos de gráficos de curvas elípticas.

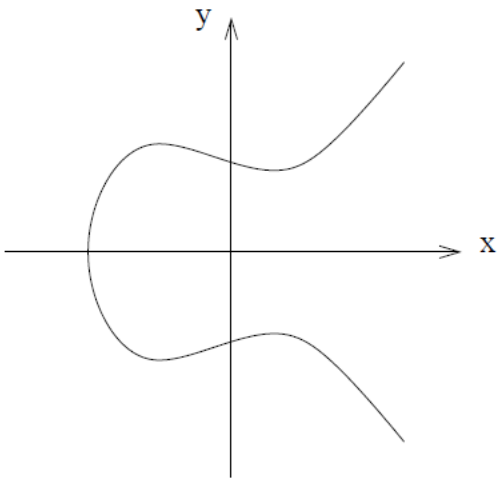

Figura 3.1: Exemplo 1

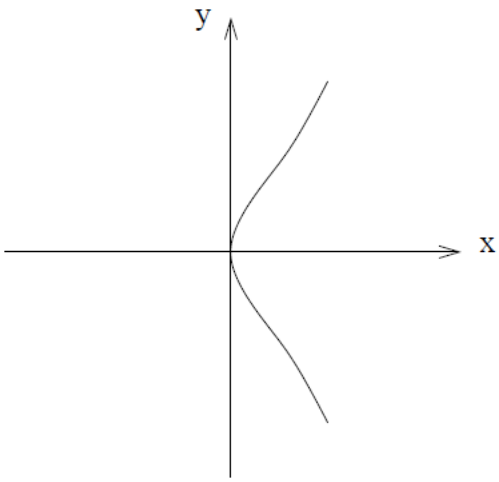

Figura 3.2: Exemplo 2

A teoria das curvas elípticas tem um fato importante que é a possibilidade de se definir uma estutura de grupo abeliano sobre essas curvas. Tomando-se dois pontos P e Q sobre uma curva elíptica qualquer, pode-se obter um terceiro ponto através de uma reta traçada sobre ela que passa por P e Q e obter-se assim o ponto de intersecção entre a reta e a curva, como mostra a Figura 3.3. Tal ponto é definido como $P * Q$.

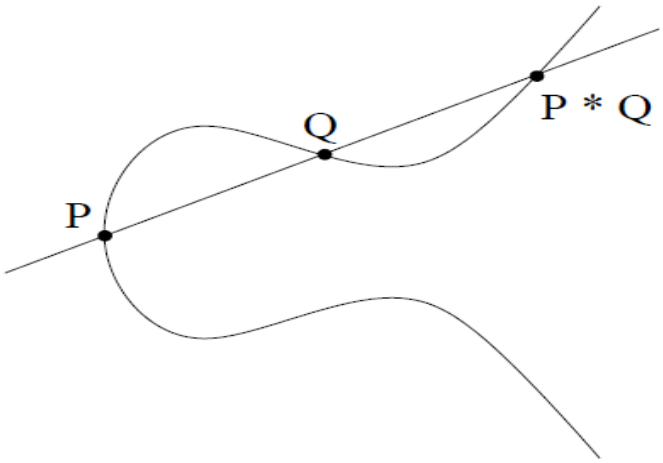

Figura 3.3: Obtenção do ponto $P * Q$

Se os pontos são iguais, considera-se a reta por P e Q como sendo a reta tangente à curva em P e obtêm-se o terceiro ponto $P * Q$, como mostra a Figura 3.4.

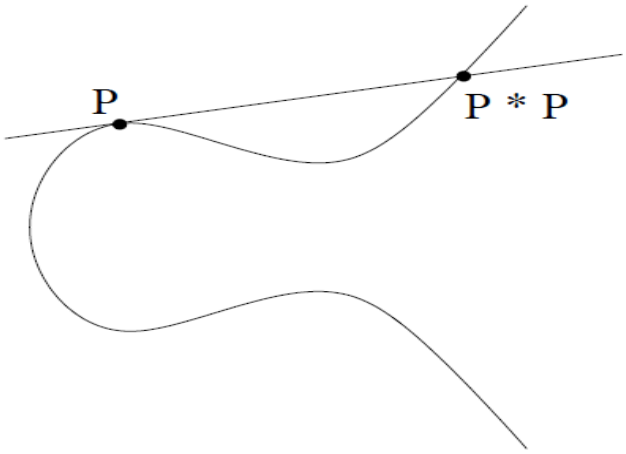

Figura 3.4: Obtenção do ponto *P * P*

3.2 Operação de adição

Seja *O* um ponto qualquer sobre *C*. A operação + toma cada par (*P*,*Q*) de pontos de *C* e agrega o ponto *P + Q* sobre *C* denotado por $P + Q = O * (P * Q)$. Assim, dados os pontos *P* e *Q*, obtém-se o terceiro ponto $P * Q$ traçando a reta L_1 que passa por *P* e *Q*; após traça-se a reta L_2 que passa por *O* e $P * Q$ e tem-se o terceiro ponto que é *P + Q*, mostrado na Figura 3.5.

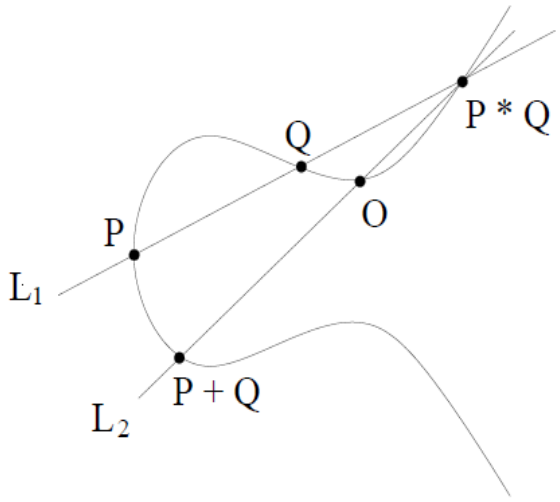

Figura 3.5: Obtenção do ponto *P + Q*

Supondo *O* como sendo o ponto no infinito, ou seja, o ponto onde as retas verticais se intersectam, pode-se adicionar dois pontos P_1 e P_2 traçando primeiramente a reta que passa por P_1 e P_2 obtendo o ponto de intersecção entre a reta e a curva. Logo após, traçamos a reta que passa por *O* e por $P_1 * P_2$ que é justamente a reta vertical que passa por $P_1 * P_2$. O ponto $P_1 + P_2$ será o ponto simétrico de $P_1 * P_2$, como mostra a Figura 3.6 a seguir:

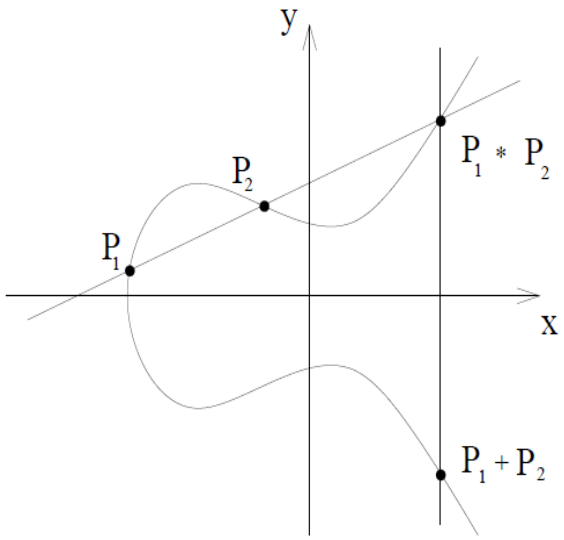

Figura 3.6: Obtenção do ponto $P + Q$ considerando-se o ponto *O* como sendo o ponto no infinito [4]

3.2.1 Propriedades da operação de adição

Associatividade: Quaisquer que sejam os pontos *P,Q e R* em *C* tem-se que
$(P + Q) + R = P + (Q + R)$;

Existência de elemento neutro: O ponto $O \in C$ é tal que $P + O = O + P = P$, $\forall P \in C$ logo *O* é o elemento neutro da adição;

Existência de elemento simétrico: Para cada ponto $P \in C$, existe o ponto $-P \in C$ tal que $P + (-P) = (-P) + P = O$.
Também vale a seguinte propriedade:
Comutatividade: Quaisquer que sejam os pontos $P,Q \in C$, tem-se $P + Q = Q + P$.

3.2.2 Adição de Pontos

Dados dois pontos $P_1 = (x_1,y_1)$ e $P_2 = (x_2,y_2)$ com $P_1 \neq P_2$ e $x_1 \neq x_2$ pertencentes à curva $C(k)$ (definida sobre um corpo K) de equação $y^2 = x^3 + ax + b$, tem-se a soma $P_1 + P_2 = (x_3,y_3)$ através das seguintes fórmulas

$$\begin{cases} \lambda = \dfrac{y_2 - y_1}{x_2 - x_1} \\ x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases} \qquad (3.4)$$

Analogicamente tem-se as fórmulas para $P_1 + P_2$ no caso em que $P_1 = P_2$;
Para a adição dos pontos P_1e P_2 com $P_1 \neq P_2$e $x_1 = x_2$, no caso em que $P_2 = - P_1$ tem-se $P_1 + P_2 = O$.

3.2.3 Duplicação de Pontos

Seja $P_1 = (x_1,y_1)$. O resultado de $2P_1 = P_1 + P_1 = (x_3,y_3)$, é obtido através das fórmulas

$$\begin{cases} \lambda = \dfrac{3x_1^{\,2} + a}{2y_1} \\ x_3 = \lambda^2 - 2x_1 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases} \qquad (3.5)$$

3.2.4 Multiplicação de Pontos

A multiplicação de um ponto P de uma curva C por um inteiro $K > 2$ é designada operação escalar de um ponto. Essa operação pode ser realizada utilizando-se as fórmulas de adição e duplicação de ponto, isto é, acrescentando o ponto a ele mesmo tantas vezes quanto for o fator de multiplicação:

$$\underbrace{KP = P + P + ... + P}_{K \text{ vezes}} \qquad (3.6)$$

3.3 Curvas elípticas sobre corpos finitos

Apesar de as curvas elípticas terem diversas aplicações nos reais, a criptografia trabalha com essas curvas definidas sobre corpos finitos. Isso acontece, em razão de problemas com arredondamento de valores e limites. Todas as aplicações desenvolvidas através da criptografia devem ser rápidas e precisas e, isso pode ser feito utilizando-se corpos finitos. Em geral, as curvas elípticas utilizadas na criptografia são definidas sobre corpos finitos primos *($\mathbb{F}_p = \mathbb{F}_p$)* ou sobre corpos finitos de característica dois ($\mathbb{F}_p = \mathbb{F}_{2^m}$). Neste trabalho será abordada a criptografia sobre curvas elípticas sobre corpos finitos primos.

Definição. *Seja p um número primo. O corpo finito $\mathbb{F}_p$ consiste do conjunto $\mathbb{F}_p = \{0,1,2...,p-1\}$, onde define-se as operações de adição e multiplicação.*

Estas operações são definidas da seguinte maneira:

Adição: Se $a,b \in \mathbb{F}_p$, então $a + b = r$, onde $r,\ 0 \leq r \geq p - 1$, é o resto da divisão de $a + b$ (adição em $\mathbb{F}$) por p. Esta operação é denominada adição módulo p.

Multiplicação: Se $a,b\ \mathbb{F}_p$, então $a \cdot b = s$, onde $r,\ 0 \leq s \geq p - 1$, é o resto da divisão de $a \cdot b$ (multiplicação em $\mathbb{F}$) por p. Esta operação é denominada multiplicação módulo p.

Uma curva elíptica C sobre $\mathbb{F}_p$ denotada por $C(\mathbb{F}_p)$ é dada pela equação da forma:

$$y^2 = x^3 + ax + b \bmod p \qquad (3.7)$$

onde $a,b \in \mathbb{F}_p$ *e* $4a^3 + 27b^2 \neq 0 \bmod p$. O conjunto $C(\mathbb{F}p)$é dado por todos os pontos (x,y), $x,y, \in Fp$ satisfazendo a Equação 3.7.

A operação de adição dos pontos $P_1 + P_2 = P_3$ onde $P_1 = (x_1,y_1)$, $P_2 = (x_2,y_2)$ *e* $P_2 = (x_3,y_3)$ em uma curva elíptica sobre $\mathbb{F}_p$ é dada por:

$$\begin{cases} \lambda = \dfrac{y_2 - y_1}{x_2 - x_1} \pmod p \\ x_3 = \lambda^2 - x_1 - x_2 \pmod p \\ \\ y_3 = \lambda(x_1 - x_3) - y_1 \pmod p \end{cases} \tag{3.8}$$

para $P_1 \neq P_2$ *e* $x_1 \neq x_2$ *mod p.*

Analogamente a operação de duplicação de um ponto dada por $2P_1 = P_1 + P_1 = (x_3,y_3)$ com $P_1 = (x_1,y_1)$ é obtida por:

$$\begin{cases} \lambda = \dfrac{3{x_1}^2 + a}{2y_1} \pmod p \\ x_3 = \lambda^2 - 2x_1 \pmod p \\ \\ y_3 = \lambda(x_1 - x_3) - y_1 \pmod p \end{cases} \tag{3.9}$$

A seguir tem-se um algoritmo que realiza a soma de pontos:

Algoritmo 1: A soma $P + Q$ em $\in C(\mathbb{F}_p)$

Entrada*:* $P(x_1, y_1)$, $Q(x_2, y_2) \in C$ e os parâmetros a,b e p da curva C.

Saída: O ponto $P + Q \in C$

início

se $P = \infty$ **então**

retorna Q;

se $Q = \infty$ **então**

retorna P;

se $P = -Q$ **então**

retorna ∞ ;

t $\leftarrow \frac{y_1 - y_2}{x_2 - x_1} \pmod p$;

x $\leftarrow t^2 - 2x \pmod p$;

$y_r \leftarrow t(x - x_r) - y \pmod p$;

retorna $(x_r, y_r) \in C$;

fim

A seguir o algoritmo de duplicação de pontos:

Algoritmo 2: Duplicação de Pontos em $C(\mathbb{F}_p)$

Entrada: Um ponto P(x,y) pertencente a curva C e os parâmetros a, b, e p da curva C

Saída: O ponto 2P $\in$C

início

se P = ∞ | mdc(2y,p) $\neq$ 1 **então**

retorna ∞;

t $\leftarrow \frac{3x^2+a}{2y}$(mod p);

$x_r \leftarrow t^2 - 2x \pmod p$;

$y_r \leftarrow t(x - x_r) - y \pmod p$;

retorna $(x_r, y_r) \in C$;

fim

Exemplo 3.1. Seja $q = 23$ a curva: $y^2 = x^3 + x + 1$, definida sobre $\mathbb{F}_{23}$, sendo $4a^3 + 27b^2 = 4 + 4 = 8 \neq 0 \bmod 23$. Os pontos, com coordenadas em $\mathbb{F}_{23}$, pertencentes a essa curva são: (0,1); (0,22); (1,7); (1,16); (3,10); (3,13); (4,0); (5,4); (5,19); (6,4); (6,19); (7,11); (7,12); (9,7); (9,16); (11,3); (11,20); (12,4); (12,19); (13,7); (13,16); (17,20); (18,3); (18,20); (19,5); (19,18) e o ponto *O*.

Exemplo 3.2. Considere a mesma curva do exemplo anterior e os pontos $P = (3,10)$ e $Q = (9,7)$ note que $P, Q \in \mathbb{F}_{23}$.

i) $P + Q = (x_3, y_3)$ em $\mathbb{F}_{23}$ é dado por:

$$\begin{cases} \lambda = \dfrac{7-10}{9-3} = \dfrac{-3}{6} = \dfrac{-1}{2} = 11 \\ x_3 = 11^2 - 3 - 9 = 6 - 3 - 9 = -6 = 17 \\ y_3 = 11(3-(-6)) - 10 = 11(9) - 10 = 20 \end{cases}$$

Portanto $P + Q = (3,10) + (9,7) = (17,20)$ em $\mathbb{F}_{23}$.

ii) $2P = P + P = (x_3, y_3)$ em $\mathbb{F}_{23}$ é dado por:

$$\begin{cases} \lambda = \dfrac{3(3)^2+1}{20} = \dfrac{5}{20} = \dfrac{1}{4} = 6 \\ x_3 = 6^2 - 6 = 7 \\ \\ y_3 = 6(3-7) - 10 = 18 + 4 - 10 = 12 \end{cases}$$

Portanto $2P = 2(3,10) = (7,12)$ em $\mathbb{F}_{23}$.

3.4 Ordem da Curva

A ordem da curva elíptica é definida como o menor inteiro positivo *n* tal que *nP = O*, sendo que *n* não precisa necessariamente existir. É importante encontrar pontos *P* de ordem finita em uma curva elíptica, principalmente para curvas definidas sobre $\mathbb{Q}$.

3.5 Teorema de Hasse

O número de pontos distintos que pertence a uma curva *#C(𝔽p)* é a ordem de uma curva elíptica. É primordial conhecer a ordem de uma curva elíptica em testes primalidade, métodos de fatoração e aplicações criptográficas que fazem o uso dessas curvas. Um relevante resultado da teoria de curvas elípticas, conhecido como Teorema de Hasse, proporciona um intervalo contendo a ordem de uma curva.

Teorema 3.1: *Seja p um número primo e C(𝔽p) uma curva elíptica sobre 𝔽p e n seu número de pontos. Então,* $|n - (q - 1)| \leq 2\sqrt{p}$. *Representando C(Fp) por #C𝔽p. O intervalo* $p + 1 - 2\sqrt{p} \leq p + 1 + 2\sqrt{p}$ *recebe o nome de intervalo de Hasse.*

Exemplo 3.3. Considerando a curva C: $y^2 = x^3 + x + 1$ definida sobre $\mathbb{F}_{23}$. Pelo Teorema de Hasse:

$$p + 1 - 2\sqrt{p} \leq \#C(\mathbb{F}_p) \leq p + 1 + 2\sqrt{p}$$

$$23 + 1 - 2\sqrt{23} \leq \#C(\mathbb{F}_{23}) \leq 23 + 1 + 2\sqrt{23}$$

$$15 \leq \#C(\mathbb{F}_{23}) \leq 34$$

Capítulo 4
Criptografia

Este capítulo relata a importância da criptografia, seus conceitos básicos e o funcionamento da mesma utilizando-se curvas elípticas.

4.1 Conceitos Básicos

Segundo Flase[8], a palavra criptografia vem do grego *cryptos* que traduzida para o português significa oculto, secreto, através da qual se estuda formas de codificar uma mensagem onde somente o destinatário tem ferramentas suficientes para interpretá-la. Por intermédio da utilização da criptografia, busca-se proteger dados existentes no computador e banco de dados. Proporciona-se segurança no processo de navegação na internet, dentre outros aspectos.

Para uma melhor compreensão Flase [8] utiliza um exemplo clássico de uma troca de mensagem, como segue:

Seja Alice o remetente da mensagem, Bob o destinatário e Eva uma invasora. A partir de um algoritmo criptográfico, Alice utiliza uma chave *K*, uma mensagem *x* e produz uma outra mensagem $y = f_k(x)$, que é a mensagem criptografada. A mensagem criptografada *y* é enviada para Bob onde *y* é descriptografada pelo algoritmo inverso $f_k^{-1}(y)$ obtendo-se *x* se, e somente se, Bob conhece a chave *K*. Supõe-se que Eva deseja decifrar a mensagem, sendo de seu conhecimento o algoritmo, mas não a chave. Dependendo da complexidade do algoritmo utilizado, torna-se praticamente impossível a decifragem da mensagem. Ou seja, Eva não pode decifrá-la e, portanto, a mensagem está protegida contra invasores.

4.2 Criptografia de chave pública ou assimétrica

O algoritmo de troca de chaves de Diffie-Hellman foi o que estabeleceu a abertura para o estudo da criptografia de chave pública, através do princípio do exemplo, a seguir, citado por Flase [8].

> Imagine que Alice deseja enviar uma mensagem pessoal e altamente secreta para Bob. Ela coloca sua carta secreta em uma caixa de ferro com um cadeado e envia para Bob. Este coloca um outro cadeado e envia para Alice novamente.

> Ao receber, Alice retira seu cadeado colocado inicialmente e reenvia para Bob, que agora pode abri-la e ler a carta, pois a caixa está trancada apenas pelo seu cadeado. Observamos aqui que é possível realizar a troca de chaves através de um canal inseguro já que todos podem saber o transporte da caixa de ferro, porém ninguém além de Alice e Bob pode descobrir o que estava escrito na carta no interior da caixa.

4.3 Criptografia baseada em Curvas Elípticas

Segundo Castellanos [10] citado por Magalhães e Querioz [9] "curvas elípticas têm sido usadas na criptografia considerando sua baixa exigência em poder computacional bem como o tamanho da chave reduzido". Assim, a justificativa fundamental para a utilização de curvas elípticas em criptografia deve-se ao fato de o problema do logaritmo discreto ser tido como intratável e, por conseguinte, tem-se a segurança do esquema. Ainda de acordo com Castellanos [10] , tem-se:

> Uma estrutura criptográfica baseada em curvas elípticas mapeia o puro texto (neologismo para palintext) em pontos de uma curva elíptica pré-estabelecida. O algoritmo segue com a parametrização desses pontos com a chave do usuário obtendo assim um segundo conjunto de pontos representativo do cifrotexto (neologismo para ciphertext). O processo de decriptação é dado pela conversão dos desses pontos (o cifrotexto) nos pontos originais com o auxílio das chaves aplicando-se assim o mapeamento inverso.

4.3.1 Multiplicação de pontos

Similarmente à multiplicação de dois elementos de uma curva elíptica sobre $\mathbb{F}_p$, tem-se a adição de dois pontos em C, onde C é uma curva elíptica sobre $\mathbb{F}_p$. Assim, a analogia de elevação para *K-ésima* potência de $\mathbb{F}_p$ é a multiplicação do ponto $P \in C$ por um inteiro K. Essa geração pode ser efetuada através da repetição do método de elevação ao quadrado em $O(logklog^3q)$ unidades de operações. Semelhantemente, a multiplicação $KP \in C$ pode ser calculada com $O(logklog^3q)$ unidades de operações pelo método da duplicação repetida [8].

Exemplo 4.1. Para encontrar 100*P*, escrevemos

$$100P = 2(50P)$$

$$100P = 2(2P + 48P)$$

$$100P = 2(2P + (2(24P))$$
$$100P = 2(2P + 2(2(12P)))$$
$$100P = 2(2P + 2(2(2(6P))))$$
$$100P = 2(2P + 2(2(2(2(3P)))))$$
$$100P = 2(2P + 2(2(2(2(2P + P)))))$$
$$100P = 2(2(P + 2(2(2(P + 2P)))))$$

e, desta maneira são efetuadas 6 duplicações e 2 adições de pontos na curva para ter-se o resultado.

A seguir é dada a proposição sobre o número de unidades de operações em uma multiplicação escalar e duas observações citadas por Flase [8]:

Proposição 4.1. *Seja uma curva elíptica C definida pela equação de Weierstrass sobre um corpo finito* $\mathbb{F}_p$. *Dado* $P \in C$, *as coordenadas de KP podem ser calculadas em* $O(\log\log^3 q)$ *unidades de operações.*

Observação 4.1. *O tempo estimado na proposição anterior não é o melhor possível, especialmente no caso em que o corpo finito tem característica p = 2, mas pode ser melhorado com as estimativas que resultam da aplicação mais conveniente da aritmética em corpos finitos.*

Observação 4.2. *Se conhecemos o número N de pontos em uma curva elíptica C e se K > N, como NP = O podemos substituir K pelo mesmo resíduo não negativo módulo N antes de calcular KP. Neste caso pode-se substituir o tempo estimado por* $O(\log^4 q)$ *(uma vez que* $N \leq q + 1 + 2\sqrt{q} = Oq$).

4.3.2 Problema do logaritmo discreto no uso de curvas elípticas

De acordo com Flase [8], o problema do logaritmo discreto aplicado na criptografia de chave pública em um corpo finito é muito conhecido. Em um grupo constituído pelos pontos de uma curva elíptica *C* definida sobre um corpo finito $\mathbb{F}_p$, tem-se a seguinte definição:

Definição 4.2. "*Sejam C uma curva elíptica sobre* $\mathbb{F}_p$ *e B um ponto de C. Então o problema do logaritmo discreto em C na base B é: Dado um ponto* $P \in C$, *deve-se encontrar um inteiro* $x \in \mathbb{F}$ *tal que xB = P, se tal inteiro existir"*.

O problema do logaritmo discreto no grupo de pontos de uma curva elíptica, em comparação com o problema do logaritmo discreto em corpos finitos, mostra-se ser mais intratável.

4.3.3 A troca de chaves Diffie-Hellman com curvas elípticas

A troca de chaves Diffie-Hellman com curvas elípticas funciona da seguinte forma: Supõe-se que Alice e Bob querem usar uma chave que será utilizada em um conjunto com um sistema criptográfico clássico. Eles precisam chegar em um acordo a respeito desta chave. Eles primeiro precisam escolher publicamente um corpo finito $\mathbb{F}_p$ e uma curva elíptica C determinada sobre ele. Através de um ponto aleatório P da curva elíptica a chave será construída. Por exemplo, eles tem um ponto aleatório $P \in C$ e, tomando a coordenada x de P obtêm-se um elemento aleatório de $\mathbb{F}_p$, que pode ser convertido em um inteiro aleatório com r dígitos na base p, onde $q = p^r$, que serve como chave para o criptossistema clássico mencionado. Deve-se ter em mente que, tem sido usado a palavra aleatório em um sentido impreciso, isto é, a escolha de P é arbitrária e imprevisível em um grande conjunto de chaves possíveis. Escolhe-se o ponto P de maneira que todas as comunicações um com o outro sejam públicas, e ninguém, além dos dois, sabe o que é P Flase[8].

Para melhor entendimento, Flase [8] utiliza o seguinte exemplo:

> Imagina-se que B é conhecido publicamente e seja um ponto fixo em C, do qual a ordem é muito grande (N ou um divisor grande de N). Aplicam-se os seguintes passos para se obter a chave P:
>
> i. Alice e Bob primeiro escolhem publicamente um ponto $B \in C$.
>
> ii. Alice escolhe um inteiro a de uma ordem de grandeza q, aleatoriamente, que é aproximadamente a mesma ordem de N, que é secreta. Ela realiza o cálculo de $abB \in C$.
>
> iii. Bob seleciona um número b aleatoriamente a qual ele o torna público $bB \in C$.
>
> iv. Alice tem o conhecimento de bB (que é público) e de seu segredo a, com isso ela pode calcular $P = abB \in C$. Bob conhece aB, e consegue calcular $P = abB \in C$.
>
> De qualquer maneira, se uma terceira pessoa tem o conhecimento de apenas aB e bB, sem resolver o problema de logaritmo discreto, não há como calcular abB.

A complexidade de tempo de execução deste algoritmo é exponencial $O(n^k)$, segundo Souza [12].

4.3.4 ElGamal sobre curvas elípticas

Segundo Lara e Oliveira [11], o algoritmo ElGamal, que trabalha com um grupo, pode-se utilizado sobre curvas elípticas. Imagina-se que Alice quer enviar uma mensagem para Bob. Seja m esta mensagem mapeada em algum ponto de uma curva elíptica. Para tanto tem-se o algoritmo:

1. Bob escolhe, e mantém em segredo, um inteiro $b \in \mathbb{N}^*$ e envia à Alice $k = bP$, sendo P um ponto da curva elíptica conhecido publicamente.
2. Alice escolhe um inteiro $a \in \mathbb{N}^*$ (também o guarda para si) e computa $c_1 = aP$ e $c_2 = m + aK$.
3. Para decifrar a mensagem m Bob calcula $c_2 - bc_1 = m + abP - baP = m$.

4.3.5 Menezes-Vanstone sobre Curvas Elípticas

De acordo com Lara [15] a técnica de Menezes-Vanstone é muito utilizada para criptografar dados utilizando curvas elípticas. Segundo esta técnica, a mensagem é um par ordenado $m = (x_1, x_2)$, com $x_1, x_2 \in \mathbb{F}_p$, com m não sendo um ponto da curva elíptica em questão. O criptograma será uma tripla ordenada $r = (y_0, y_1, y_2)$, onde $y_1, y_2 \in \mathbb{F}_p$ e y_0 é um ponto da curva elíptica.

O algoritmo utilizado por esse método é descrito da seguinte forma:

Para criptografar $m = (x_1, x_2)$;

1. Bob escolhe um inteiro $K \in \mathbb{F}_p$ e calcula $y_0 = KP$ (lembre-se que Bob conhece publicamente o ponto $P \in C$).
2. Bob computa $(c_1, c_2) = KQ$, $y_1 = c_1x_1 \bmod p$ e $y_2 = c_2x_2 \bmod p$. Envia para Alice a tripla $r = (y_0, y_1, y_2)$.

Para descriptografar $r = (y_0, y_1, y_2)$;

1. Alice calcula $sy_0 = sKP = KQ = (c_1, c_2)$, onde s é um inteiro selecionado por Alice. Observe que $Q = sP$.
2. Logo após, Alice calcula $x_1 = y_1 (c_1)^{-1} \bmod p$ e $x_2 = y_2 (c_2)^{-1} \bmod p$, recuperando a mensagem $m = (x_1, x_2)$.

Exemplo 4.1: A Tabela 4.1 contém as letras do alfabeto com o seus respectivos números que serão utilizados no processo de cifragem e decifragem.

Tabela 4.1

A	B	C	D	E	F	G	H	I	J	K	L	M	
01	02	03	04	05	06	07	08	09	10	11	12	13	
N	O	P	Q	R	S	T	U	V	X	W	Y	Z	
14	15	16	17	18	19	20	21	22	23	24	25	26	27

Curva: $y^2 = x^3 + 373x + 402$ sobre $\mathbb{F}_{3697}$ com o ponto P = (551,1946)

Alice vai enviar a palavra UEMS.

UEMS = 21, 05, 13, 19 =(21, 05), (13,19) = $(x_1, x_2), (x_3, x_4)$

A cifragem ocorre da seguinte forma:

1. Bob escolhe sua chave secreta B = 815;
2. Bob calcula BP = 815(551,1946) = (958,14) e envia para Alice;
3. Alice escolhe sua chave secreta A = 919;
4. Alice calcula AP = 919(551,1946) = (301,3454);
5. Alice calcula a chave compartilhada = $AP * B$ = 919(958,14) = (837,2461) = (c_1, c_2)
6. Alice calcula:

$$c_1 * x_1 = 837 * 21 \bmod 3697 = 2789 = y_1$$

$$c_2 * x_2 = 2461 * 05 \bmod 3697 = 1214 = y_2$$

$$c_1 * x_3 = 837 * 13 \bmod 3697 = 34\,87 = y_3$$

$$c_2 * x_4 = 2461 * 19 \bmod 3697 = 2395 = y_4$$

Alice envia (AP, y_1, y_2, y_3, y_4).

Decifragem

Bob calcula $B * AP$ = 919(958,14) = (837,2461) = (c_1, c_2)

$x_1 = (y_1 * c_1^{-1}) \bmod 3697 = (2789 * 837^{-1}) \bmod 3697 = (2786 * 2447) \bmod 3697 = 21$

$x_2 = (y_2 * c_2^{-1}) \bmod 3697 = (1214 * 2461^{-1}) \bmod 3697 = (2789 * 1008) \bmod 3697 = 05$

$x_3 = (y_3 * c_1^{-1}) \bmod 3697 = (3487 * 837^{-1}) \bmod 3697 = (2789 * 2447) \bmod 3697 = 13$

$x_4 = (y_4 * c_2^{-1}) \bmod 3697 = (2397 * 2461^{-1}) \bmod 3697 = (2789 * 1008) \bmod 3697 = 19$

Pela tabela de pré-codificação 21 = U, 05 = E, 13 = M, 19 = S.

Capítulo 5

Implementação

Esse capítulo apresenta uma descrição dos aspectos de implementação da criptografia de curvas elípticas. Para tanto, foi utilizado o algoritmo de Menezes-Vantene.

5.1 Ambiente de Desenvolvimento

Os algoritmos foram implementados na linguagem C, utilizando o compilador GCC e o sistema operacional Windows7.

Todos os testes apresentados neste trabalho foram realizados em um computador com processador Core i3 de 2.13 GHz e 3 GB de memória RAM.

5.2 Implementação

Para a realização deste trabalho foram implementados três algoritmos principais, um para criar as chaves de codificação e decodificação, um para codificar e outro para decodificar uma mensagem.

O algoritmo que cria as chaves tem como entrada três inteiros, *x, y* e *A*, sendo que, *x e y* representam um ponto *P*, e o inteiro *A* representa a chave secreta de Alice ou Bob. Com estes dados e utilizando uma curva elíptica pré-definida é possível calcular uma chave pública.

O algoritmo que codifica uma mensagem tem como entrada dois pontos *P e BP*, a chave secreta de quem quer enviar a mensagem e um arquivo contendo a mensagem a ser enviada. Primeiramente o algoritmo realiza a pré-codificação de acordo com a Tabela 4.1. Em seguida ele realiza a codificação segundo o método Menezes-Vanstone, visto na Seção 4.3.5.

O algoritmo que decodifica a mensagem obtida através do algoritmo de codificação, tem como entrada um ponto *P(* chave secreta de quem quer decodificar a mensagem) e um arquivo contendo um ponto *AP* (chave pública de quem enviou a mensagem) e a mensagem codificada.

5.2.1 Algoritmo de codificação

Na fase de codificação foi definida a curva elíptica $x^3 + 373x + 402$ e o ponto público *P*= (551,1946). Foram criptografadas com esse método de criptografia duas mensagens: a primeira

“UEMS” e a segunda “Tiago estuda com a Geisiane”. A troca de chaves foi realizada através do algoritmo de troca de chaves, sendo que a chave secreta do emissor foi 919, e a do receptor foi 815.

Para realizar a codificação, o programa criado pede ao usuário o nome do arquivo que contém a mensagem a ser criptografada. Após a confirmação de que o nome digitado é de um arquivo, o programa pré-codifica a mensagem contida no mesmo, seguindo a tabela de pré-codificação citada na seção 4.3.5.

Utilizando o método de troca de chaves de Diffie Hellman explicado na Seção 4.3.3, tanto o emissor como o receptor compartilham uma chave. Com a chave compartilhada é feito o processo de criptografia, explicado na seção 4.3.5 obtendo desta forma a mensagem criptografada. Depois de todo esse processo o programa grava em um arquivo, com nome saída.txt, a mensagem criptografada.

5.2.2 Algoritmo inversor

No processo de decodificação é necessária a utilização de um algoritmo inversor, algoritmo este que tem como entrada dois inteiros, *num* e *z*, onde *num* representa o número a ser invertido e *z* representa o corpo finito em que a curva está definida.

Este algoritmo faz várias verificações para descobrir qual elemento é o inverso de *num* módulo *z*. Ao descobrir qual é o elemento inverso de *num*, este algoritmo o retorna. Esse algoritmo tem complexidade *O(z).*

5.2.3 Algoritmo de decodificação

Como na fase de codificação, na decodificação foi definida a curva elíptica $x^3 +373x+402$ e o ponto público P = (551,1946). Foram decodificadas com esse método as duas mensagens, obtidas através do método de codificação.

Para realizar a decodificação o programa criado pede ao usuário o nome do arquivo que contém a mensagem a ser decodificada. Após a confirmação de que o nome digitado é de um arquivo, o programa guarda a mensagem codificada.

Tendo a mensagem pré-codificada, o programa pede ao usuário que digite a chave compartilhada entre o emissor e receptor. Com essa chave faz-se o processo de decodificação,

explicado na seção 4.3.5, obtendo desta forma a mensagem original. Depois de todo esse processo o programa grava em um arquivo, com nome saída.txt, a mensagem decodificada.

Capítulo 6

Testes

Neste capítulo são apresentados alguns testes realizados com os algoritmos de codificação e decodificação. Foram efetuados testes com uma mesma mensagem para curvas elípticas diferentes, bem como testes de mensagens de tamanhos diferentes para análise de tempo de execução.

6.1 Testes com trocas de curvas

6.1.1. Primeira Curva: $y^2 = x^3 + 954x + 2318$

O corpo finito escolhido para testar os algoritmos nesta curva foi $\mathbb{F}_{3697}$ e ponto público escolhido para esta curva foi P = (212,2453). Desta forma, a chave compartilhada foi TP = (974,2144).

A mensagem obtida no processo de codificação foi a seguinte:

995 811 974 220 3519 2433 1173 69 995 660 199 2144 419 2735 3519 1993 419 2144 419 220 1173 811 21 3326

No processo de decodificação, tendo como entrada a mensagem citada acima, foi o obtida a seguinte mensagem:

TIAGO ESTUDA COM A GEISE

6.1.2. Segunda Curva: $y^2 = x^3 + 26x + 4385$

O corpo finito escolhido para testar os algoritmos nesta curva foi $\mathbb{F}_{5297}$ e o ponto público escolhido para esta curva foi P = (563,3544). Desta forma a chave compartilhada foi TP = (1023,3244).

A mensagem obtida no processo de codificação foi a seguinte:

4569 2711 1023 1520 4751 2836 5115 3369 4569 4560 4092 3244 1136 4435 4751 5093 1136 3244 1136 1520 5115 2711 3546 329

O processo de decodificação, retornou a mensagem de entrada.

6.1.3. Terceira Curva: $y^2 = x^3 + 954\,x + 1659$

O corpo finito escolhido para testar os algoritmos nesta curva foi $\mathbb{F}_{3631}$ e ponto público escolhido para esta curva foi P = (2053,2356). Desta forma a chave compartilhada foi TP = (351,480).

As mensagens enviadas para o processo de cifragem nesta curva foram cinco mensagens randômicas a primeira com 100 caracteres, a segunda com 1000 caracteres, a terceira com 20000 caracteres, a quarta com 100000 caracteres e a quinta com 300000 caracteres.

As mensagens obtidas no processo de codificação foram enviadas para o processo de decodificação.

O processo de decodificação retornou as mensagens randômicas originais.

6.2 Testes de mensagens com diferentes quantidades de caracteres

Foram testadas mensagens com diferentes quantidades de caracteres. A Figura 6.1 apresenta os tempos de execução obtidos nos processos de codificação e decodificação.

Pode-se observar que, nos dois processos, o tempo de execução aumenta consideravelmente à medida em que é acrescida a quantidade de caracteres nas mensagens.

Também pode-se constatar que o processo de decodificação apresenta um tempo maior de execução em comparação ao processo de codificação.

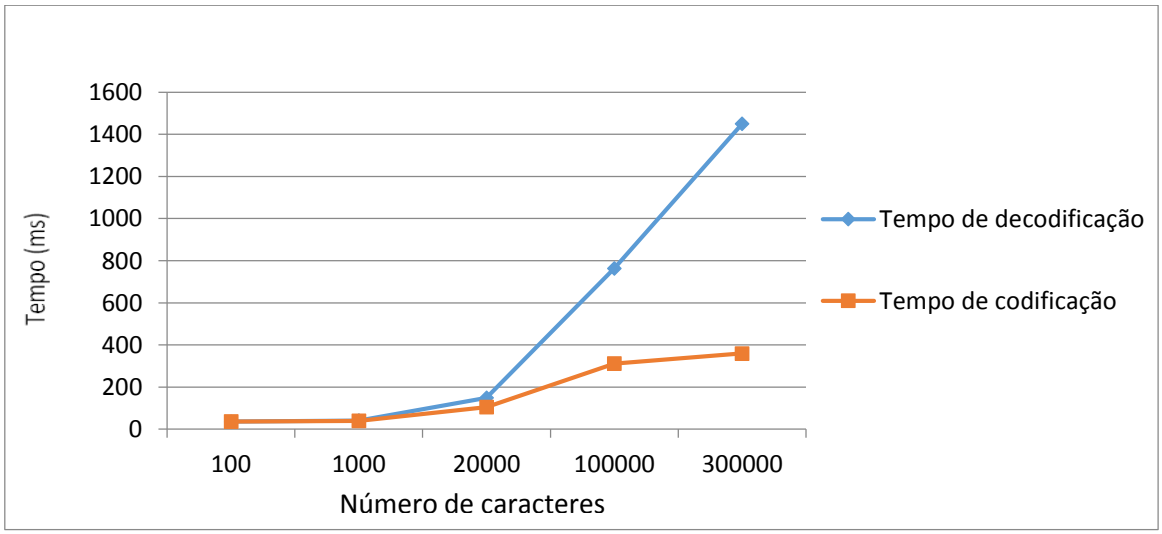

Figura 6.1: Comparativo de tempo de codificação e decodificação

Capítulo 7

Considerações Finais

Desde os séculos passados, busca-se a segurança na troca de informações e dados. Nesse sentido, procura-se estudar métodos criptográficos que possam realizar a codificação e a decodificação das informações, de forma segura. Dentre os métodos criptográficos existentes está a criptografia de curvas elípticas.

Esse trabalho apresentou o funcionamento do método em questão, bem como alguns conceitos matemáticos e de curvas elípticas, fundamentais para sua compreensão. Também foram mostrados resultados de testes realizados com uma implementação desse sistema criptográfico, utilizando três curvas elípticas, pontos e corpos finitos diferentes.

Para trabalhos futuros recomenda-se um estudo mais aprofundado dos algoritmos aplicados no funcionamento do método em questão, bem como a realização de testes para uma análise de seu desempenho e estudos comparativos com outros métodos criptográficos.

Apêndice A

Multiplicação de pontos

```
pontos Multiblica2P(pontos P, int a , int z)
   {
    int k, c, aux, aux2;
    int auxx;
    pontos P2;
    if(((P.x==0)&&(P.y==0))||(mdc((2*P.y),z)!=1))
      {
       return(P);
      }
    P2.x=0;
    P2.y=0;
    aux=0;
    k= P.x * P.x;
    k=3*k;
    k=k+a;
    c=2*P.y;
    k=(k*(inversor(c,z)))%z;
    aux = k*k;
    aux=aux-(2*P.x);
    P2.x=aux%z;
    aux2=(-P.y+(k*(P.x-P2.x)));
    if(aux2<0)
      {
       aux2=aux2*-1;
      }
    P2.y=aux2%z;
    return (P2);
   }
```

Apêndice B

Soma de pontos

```
pontos SomaPontos( pontos P1, pontos P2,int z)
   {
    pontos P3;
    int k,auxx;
    int l,k1, aux, aux2;
    k=0;
    aux=0;
    k=P2.y - P1.y;
    aux =(P2.x - P1.x);
    aux=aux%z;
    k=(k*(inversor(aux,z)))%z;
    aux=(k1*k1)-P1.x - P2.x;
    if(aux<0)
      {
       aux=aux*-1;
      }
    P3.x=aux%z;
    aux2=(P1.x - P3.x);
    aux2= aux2*k;
    aux2=(aux2-P1.y);
    if(aux2<0)
      {
       aux2=aux2*-1;
      }
    P3.y=aux2%z;
    return(P3);
   }
```

Apêndice C

Decodificação

```
void Decodifica(int vet[],int tamanho,int a, int z,int B,pontos AP)
   {
    int i,j;
    pontos TP;
    char v[50]="  ";
    TP=MultPontos(B, a, AP,z);
    for(i=0;i<tamanho;i+=2)
       {
        vet[i]=(vet[i]*(inversor(TP.x,z))) %z;
        vet[i+1]=(vet[i+1]*(inversor(TP.y,z)))%z;
       }
      j=i;
    for(i=0;i<tamanho;i++)
       {
        v[i]=contrucaoTabelaDecodificacao(vet[i]);
       }
    GravaArquivoSaida(j,v);
   }
```

Apêndice D

Codificação

```
void Codifica(int vet[],int tamanho,int a, int z,pontos BP,int A)
   {
    int i;
    pontos TP;
    TP=MultPontos(A, a, BP,z);
    for(i=0;i<tamanho;i+=2)
      {
       vet[i]=(vet[i]*TP.x) %z;
       vet[i+1]=(vet[i+1]*TP.y) %z;
      }
    GravaArquivoSaida(tamanho,vet);
   }
```

Referências Bibliográficas

1. Medeiros, D. A. *Utilização de Curvas Elípticas na Fatoração de Números Inteiros.* Trabalho de Conclusão de Curso, Universidade Estadual do Mato Grosso do Sul, 2010.
2. Dias, R. A. R. *Algumas Evidências Computacionais da Infinitude dos Números Primos de Fibonacci e Generalizações Destes*. Monografia, Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte, 2008.
3. Marques, C. M. *Introdução à Teoria de Anéis*. Departamento de Matemática, Universidade Federal de Minas Gerais, 1999 com revisão em 2005 . Disponível em: http://www.mat.ufmg.br/~marques/Apostila-Aneis.pdf
4. Molgora, A. P. *Uma implementação do Método das Curvas Elíticas para Fatoração de números Inteiros*. Dissertação de Mestrado, Departamento de Computação e Estatística, Universidade Federal de Mato Grosso do Sul, 2006.
5. Barbosa, J. C. *Criptografia de Chave Pública Baseada em Curvas Elípticas*. Monografia final de curso, Universidade Federal do Rio de Janeiro , 2003 disponível em: http://www.lockabit.coppe.ufrj.br/sites/lockabit.coppe.ufrj.br/files/publicacoes/lockabit/eccmono.pdf
6. Nascimento, P. G. S. *Criptografia de Chave Pública Baseada em Curvas Elípticas*, Rio de Janeiro, Disponível em: http://artigos.netsaber.com.br/resumo_artigo_36419/artigo_sobre_criptografia_de_chave_p%C3%9Ablica_baseada_em_curvas_el%C3%8Dpticas
7. Dantas, D. A. M. e Rodriguez, J. E. A. *Curvas Elípticas e o Teorema de Hasse Ellíptic Curves and Hasse's Theorem,* São Paulo.Universidade Estadual Paulista, n.2. Disponível em: http://prope.unesp.br/xxi_cic/27_06868023650.pdf
8. Flase, V. B. A. S. *Criptografia e Curvas Elípticas. Dissertação de Mestrado*. Instituto de Deociências e Ciências Exatas. Universidade Estadual Paulista, 2011.
9. Magalhães, D. K. S. e Queiroz, J. G. B.. *Curvas Elípticas aplicadas á Criptografia.* Artigo, Centro de Informática, Universidade Federal do Pernambuco, 2001. Disponível em: http://www.die.ufpi.br/ercemapi2011/artigos/ST2_07.pdf
10. Castellanos, A.S. *Criptografia usando curvas hiperelípticas*. Universidade Estaudal de Campinas, 2004.

11. Lara, P. C. S. e De Oliveira, F. B. *Curvas Elípticas: Aplicação em Criptografia Assimétrica*. Artigo, Coordenação de Sistemas e Redes, Laboratório Nacional de Computação Científica.
12. De Souza, R. C. *Criptografia de Chave Pública: Algoritmos que possibilitam a criação de chave Assimétrica*. Artigo. Disponível em: http://www.ucb.br/sites/100/103/TCC/22005/RaimundoCandidodeSousa.pdf
13. Sangalli, L. A. *Criptossistemas baseados em curvas elípticas e seus desafios*. Departamento de Engenharia de Computação e Automação Industrial. Universidade Estadual de Campinhas (UNICAMP), Disponível em: http://www.dca.fee.unicamp.br/portugues/pesquisa/seminarios/2012/artigos/217.pdf
14. Quaresma, P. e Lopes, Elsa. *Criptografia. Departamente de Matemática*, Universidade de Coimbra. Disponível em: http://www.mat.uc.pt/~pedro/lectivos/CodigosCriptografia1011/artigo-gazeta08.pdf
15. Lara, P.C.S. *Implementação de uma API para Criptografia Assimétrica Baseada em Curvas Elípticas*. Instituto Superior de Tecnologia em Ciência da Computação de Petrópolis. Fundação de Apoio à Escola Técnica do Estado do Rio de Janeiro – FAETEC, 2008.