UNIVERSIDADE ESTADUAL DE LONDRINA

ADRIANO GOMES DE SANTANA

CRIPTOGRAFIA DE CURVAS ELÍPTICAS SOBRE EXTENSÕES DE CORPOS FINITOS

Londrina
2013

ADRIANO GOMES DE SANTANA

CRIPTOGRAFIA DE CURVAS ELÍPTICAS SOBRE EXTENSÕES DE CORPOS FINITOS

Dissertação de mestrado apresentada ao Departamento de Matemática da Universidade Estadual de Londrina, como requisito parcial para a obtenção do Título de MESTRE em Matemática Aplicada e Computacional.

Orientador: Prof. Dr. Naresh Kumar Sharma

Londrina
2013

Catalogação elaborada pela Divisão de Processos Técnicos da Biblioteca Central da Universidade Estadual de Londrina

Dados Internacionais de Catalogação -na-Publicação (CIP)

S232c Santana, Adriano Gomes de.
Criptografia de curvas elípticas sobre extensões de corpos finitos / Adriano Gomes de Santana. – Londrina, 2012.
74 f. : il.

Orientador: Naresh Kumar Sharma.
Dissertação (Mestrado em Matemática Aplicada e Computacional) – Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Matemática Aplicada e Computacional, 2012.

Inclui Bibliografia.

1. Computação – Matemática aplicada – Teses. 2. Criptografia – Teses. 3. Curvas elípticas – Teses. 4. Anéis de endomorfismo – Teses. 5. Corpos finitos (Álgebra) – Teses. 6. Polinômios – Teses. I. Sharma, Naresh Kumar. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Matemática Aplicada e Computacional. III. Título.

519.681-7

ADRIANO GOMES DE SANTANA

CRIPTOGRAFIA DE CURVAS ELÍPTICAS SOBRE EXTENSÕES DE CORPOS FINITOS

Dissertação de mestrado apresentada ao Departamento de Matemática da Universidade Estadual de Londrina, como requisito parcial para a obtenção do Título de MESTRE em Matemática Aplicada e Computacional.

BANCA EXAMINADORA

Prof. Dr. Naresh Kumar Sharma
Universidade Estadual de Londrina

Prof. Dr. Mauri Cunha do Nascimento
Universidade do Estado de São Paulo

Prof[a]. Dr[a]. Ana Lúcia da Silva
Universidade Estadual de Lodrina

Londrina, 06 de FEVEREIRO de 2013.

Dedico este trabalho a Aline da Costa Venancio

AGRADECIMENTOS

Ao Deus onipresente, onisciente e onipotente que a mim nunca abandou e foi provisão naquilo que minha mão não alcançava.

Ao meu orientador prof. Dr. Naresh K. Sharma pelos conselhos, sabedoria, amizade e inspiração.

Á minha esposa Aline C. Venancio pela paciência, carinho, atenção, incentivos e companheirismos em todo o percurso do mestrado.

Aos meus pais Valdemar B. Santana e Eva G. Santana pelas ajudas, apoios e confianças.

Aos professores e colegas do programa por me ajudarem a chegar até aqui.

À CAPES pelo apoio financeiro.

Meus sinceros agradecimentos.

“O pensamento é apenas um lampejo entre duas longas noites, mas esse lampejo é tudo”

Henri Poincaré

SANTANA, Adriano Gomes de. **Criptografia de Curvas Elípticas Sobre Extensões de Corpos Finitos**. 2013. Número total de folhas. Dissertação (Mestrado em Matemática Aplicada e Computacional) – Universidade Estadual de Londrina, Londrina, 2013.

RESUMO

Um sistema de criptografia de curvas elípticas se baseia no uso do algoritmo de criptografia de chave pública de ElGamal sobre o grupo de pontos de uma curva elíptica definida sobre um corpo finito. Em geral, os protocolos de segurança para computadores utilizam apenas curvas elípticas definidas sobre corpos de cardinalidade prima p ou 2^k. Neste trabalho é proposto o uso do grupo de pontos em extensões finitas do corpo de definição de uma curva elíptica; para isso é desenvolvido um algoritmo de adição de pontos utilizando o endomorfismo de Frobenius que, em certa classe de curvas, é mais eficiente que o algoritmo tradicional. Também é descrito um método eficiente para obter a ordem do grupo de pontos destas curvas. Finalmente é apresentado uma generalização do algoritmo de primalidade de Miller para a obtenção de polinômios irredutível sobre corpos finitos, essenciais para o trabalho com extensões destes corpos, e os resultados obtidos a partir da implementação destes algoritmos.

Palavras-chave: Criptografia. Curvas Elípticas. Endomorfismo. Corpos Finitos. Polinômios Irredutíveis.

SANTANA, Adriano Gomes de. **Elliptic Curves Cryptography on Extensions of Finite Fields**. 2013. Número total de folhas. Dissertação (Mestrado em Matemática Aplicada e Computacional) – Universidade Estadual de Londrina, Londrina, 2013.

ABSTRACT

An elliptic curve cryptosystem is based on the use of the encryption algorithm of public key of ElGamal on the group of points of the elliptic curve over a finite field. In general, the security protocols for computers use only elliptic curves defined over fields of cardinality prime p or 2^k. In this work, is proposed the use of the group of points in finite extensions of the field of definition of the elliptic curve; for this an algorithm of addition of points using the endomorphism of Frobenius, which is more efficient than the traditional algorithm to a certain family of curves, is developed. An eficient method to obtain the order of the group of points of these curves is also described. Finally, a generalization of the Miller's algorithm of primality is given to obtain irreducible polynomals over finite fields, necessary to work with extensions of these fields, and the results obtaind based on implementation of these algorithms.

Keywords: criptography. elliptic curve. endomorphism. finite fields. irreducible polynomials.

LISTA DE FIGURAS

4.1 Curvas elípticas sobre $\mathbb{R}$ 36
4.2 Adição de pontos em curvas elípticas não-singulares sobre $\mathbb{R}$ 38

5.1 Tempo percentual teórico gasto pelo algoritmo 5.5 com relação ao algoritmo 2.6 56

6.1 Diagramas de dispersão e retas de regressão linear para os dados da tabela 6.1 . 65
6.2 Tempo percentual gasto para uma chave de tamanho de 64 bits 67
6.3 Tempo percentual gasto para uma chave de tamanho de 128 bits 67
6.4 Tempo percentual gasto para uma chave de tamanho de 256 bits 68

LISTA DE TABELAS

2.1 ASCII (American Standard Code for Information Interchange) 18
2.2 Algoritmo de Criptografia de ElGamal 22

4.1 Algoritmo de Criptografia de Curvas Elípticas 48

6.1 Número médio de tentativas para encontrar um polinômio de grau k sobre $\mathbb{Z}_p[X]$ 64
6.2 Tempo necessário para obter um polinômio irredutível em $\mathbb{Z}_p$ dado o tamanho da chave 66

LISTA DE ABREVIATURAS E SIGLAS

DHP	Diffie-Hellman Problem.
DLP	Discrete Logarithm Problem.
ECDLP	Elliptic Curve Discrete Logarithm Problem
SSL	Secure Sockets Layer.
ASCII	American Standard Code for Information Interchange.
DIP	Domínio de Ideais Principais.
MOV	Menezes Okamoto Vastones (algoritmo).

LISTA DE SÍMBOLOS E NOTAÇÕES[1]

$\mathbb{N}$	Conjunto dos números Naturais.
$\mathbb{Z}$	Conjunto dos números Inteiros.
$\mathbb{R}$	Conjunto dos números Reais.
$\mathbb{C}$	Conjunto dos números Complexos.
$(G, \circ)$	Grupo com composição $\circ$.
$O(g(n))$	Custo assintótico
$\lfloor x \rfloor$	Maior inteiro menor ou igual a x
$\mathbf{R}^*$	Conjunto dos elementos não-nulos de $\mathbf{R}$
$(\mathbf{R}, +, \cdot)$	Anel
$\mathbf{K}\|\mathbf{F}$	$\mathbf{K}$ é uma extensão de corpos de $\mathbf{F}$
$\cong$	Isomorfismo
(S)	Ideal gerado pelo conjunto S
(a)	Ideal gerado pelo conjunto $\{a\}$
$a \equiv b \mod \mathbf{I}$	Relação de congruência módulo $\mathbf{I}$
$\mathbf{cl}(a)$	Classe de equivalência de a
$\frac{\mathbf{R}}{\mathbf{I}}$	Anel quociente de $\mathbf{R}$ por $\mathbf{I}$
$\mathbb{Z}_n$	Anel quociente de $\mathbb{Z}$ por $(n) = n\mathbb{Z}$
$\mathbf{car}(\mathbf{F})$	Característica de $\mathbf{F}$
$f(X)$	Polinômio
$\mathbf{gr}(f)$	Grau do polinômio f
$\mathbf{R}[X]$	Conjunto dos polinômios com coeficientes em $\mathbf{R}$
$a\|b$	a divide b
$[\mathbf{K} : \mathbf{F}]$	Grau de extensão de $\mathbf{K}$ sobre $\mathbf{F}$
$\mathbb{F}_q$	Corpo finito com q elementos
$E_{\mathbf{F}}(\mathbf{K})$	Curva elíptica sobre $\mathbf{K}$ com coeficientes em $\mathbf{F}$
$E(\mathbf{F})$	Curvas elíptica sobre $\mathbf{F}$
$\Delta(E)$	Discriminante de E
$j(E)$	j-invariante de E
$\mathcal{M}$	Custo da multiplicação num corpo $\mathbf{F}$
$\mathcal{I}$	Custo da inversão num corpo $\mathbf{F}$

[1] As notações e símbolos usuais não são listados.

$\#(A)$	Cardinalidade do conjunto A
$End(E)$	Conjunto dos endomorfismo de E
ϕ_q	Endomorfismo de Frobenius
$(x : y : z)$	Ponto do plano projetivo
$E[n]$	Conjunto dos pontos de n-torção
$e(P, Q)$	Emparelhamento de Weil

SUMÁRIO

1 INTRODUÇÃO **15**

2 CRIPTOGRAFIA, ELGAMAL E LOGARITMO DISCRETO **17**

2.1 CRIPTOGRAFIA SIMÉTRICA E ASSIMÉTRICA 17

2.2 ALGORITMO DE CRIPTOGRAFIA DE ELGAMAL 19

2.3 PROBLEMA DE DIFFIE-HELLMAN E PROBLEMA DO LOGARITMO DISCRETO 23

3 CORPOS **27**

3.1 ANÉIS E CORPOS 27

3.2 EXTENSÃO DE CORPOS 31

3.3 CORPOS FINITOS 33

4 CURVAS ELÍPTICAS **35**

4.1 ADIÇÃO DE PONTOS 37

4.2 ENDOMORFISMO DE FROBENIUS 39

4.3 CURVAS ELÍPTICAS SOBRE CORPOS DE CARACTERÍSTICA 2, 3 E DIFERENTE DE 2 E 3 41

4.4 SISTEMAS DE COORDENADAS 43

4.4.1 **Plano projetivo** 43

4.4.2 **Coordenadas Jacobianas** 45

4.4.3 **Coordenadas de Edwards** 46

4.5 CRIPTOGRAFIA DE CURVA ELÍPTICAS 47

4.6 ALGORITMO MOV 48

5 CURVAS ELÍPTICAS SOBRE $\mathbb{F}_{q^k}$ COM COEFICIENTES EM $\mathbb{F}_q$ **51**

5.1 ORDEM DOS PONTOS DE $E(\mathbb{F}_{q^k})$ 51

5.2 MULTIPLICAÇÃO POR ESCALAR INTEIRO 52

5.3 CURVAS DE KOBLITZ 56

5.4 POLINÔMIOS IRREDUTÍVEIS 57

6 RESULTADOS PRÁTICOS **64**

6.1 POLINÔMIOS 64

6.2 MULTIPLICAÇÃO DE UM PONTO POR UM INTEIRO 66

7 CONCLUSÃO **69**

REFERÊNCIAS **71**

1 INTRODUÇÃO

Os avanços tecnológicos conquistados até hoje possibilitou que muitas das atividades diárias como compras, pagamento de contas, transferências bancárias e recebimentos pudessem ser feitos por meio de um computador com acesso à internet. Como as informações da internet navegam por fios ou por ondas de rádio, tais atividades não seriam possível se não fosse o desenvolvimento de algoritmos de criptografia de chave pública, estes por sua vez mantem em sigilo informações que mesmo interceptadas não podem ser usadas em práticas ilegais.

Além de aplicações na criptografia, o estudo de curvas elípticas possibilitou a descoberta de muitos outros resultados obtidos na Matemática. Por exemplo, o "último teorema de Fermat" é ele próprio um corolário da conjectura de Taniyama-Shimura demonstrado por Andrews Wiles [22] em 1995 que versa sobre uma relação entre formas modulares e curvas elípticas. Antes disso, casos particulares do mesmo teorema já haviam sido demonstrados utilizando tais curvas.

A utilização do algoritmo de criptografia de chaves públicas de Taher ElGamal [7] no grupo dos pontos de uma curva elíptica sobre um corpo finito é o que constitui a criptografia de curvas elípticas. A segurança do algoritmo de ElGamal se encontra na dificuldade de resolver o problema de logaritmo discreto (DLP - Discrete Logarithm Problem), que possui um custo subexponencial. Entretanto, o problema do logaritmo discreto para curvas elípticas (ECDLP - Elliptic Curve Discrete Logarithm Problem) possui um custo exponencial para ser resolvido, o que é uma grande vantagem com relação ao caso geral.

Embora exponencial, esta propriedade para a segurança de um sistema de criptografia com curvas elípticas só é obtido levando em considerações algumas propriedades. Por exemplo, a ordem do grupo dos pontos de uma curvas elíptica não pode ser pequena ou ter uma fatoração em primos pequenos, caso em que o ECDLP pode ser resolvido a um custo polinomial. Outra questão de segurança é evitar o uso de curvas supersingulares em que o ECDLP pode ser resolvido a um tempo subexponencial.

Os protocolos de criptografia de curvas elíptica atuais utilizam apenas corpos finitos com cardinalidade prima p ou 2^k. Isto é o caso do protocolo de criptografia SSL (Secure Sockets Layer) em sua versão 10.0 [13]. O fato de se usarem corpos com cardinalidade prima p é devido a sua fácil representação pelos elementos de $\mathbb{Z}_p$, já a utilização de corpos cuja cardinalidade seja 2^k decorre das curvas de Koblitz [4] definidas sobre estes corpos.

Este trabalho propõem a utilização da criptografia de curvas elípticas sobre extensões de corpos finitos com característica pequena. Nestas curvas pode-se usar o endomorfismo de Frobenius para calcular de maneira eficiente a codificação e decodificação de mensagens. Além disso, o teorema de Hasse [18, 21] indica um método para obter a ordem do grupo dos pontos destas curvas elípticas eficientemente. Conhecer esta ordem é imprescindível para saber se a curvas elíptica é ou não segura na criptografia.

No segundo capítulo deste trabalho é tratado sobre os aspectos básicas de um algoritmo de criptografia de chave pública, o algoritmo de ElGamal e uma análise geral do problema do logaritmo discreto.

No terceiro capítulo é tradada a estrutura de anéis e corpos, extensão de corpos e a representação dos elementos de extensões de corpos finitos.

No quarto capítulo aborda-se conceitos de curvas elíptica como: a equação de Weierstrass, singularidade, isomorfismos, adição de pontos e o endomorfismo de Frobenius. Também apresenta-se o algoritmo de criptografia para curvas elípticas, a redução do ECDLP para um DLP sobre um corpo finito utilizando o algoritmo MOV [17, 11] e curvas super-singulares.

O quinto capítulo trata da utilização de extensões de curvas elípticas sobre corpos finitos de característica pequena, os algoritmos para o cálculo da ordem dos pontos, a adição de pontos sobre estas curvas e as curvas de Koblitz. A última seção deste capítulo trata da obtenção de polinômios irredutíveis, necessários para representar e trabalhar corpos finitos.

O último capítulo apresenta resultados obtido pela implementação dos algoritmos descritos durante o trabalho, a comparação com algoritmos atuais e a viabilidade dos novos algoritmos.

2 CRIPTOGRAFIA, ELGAMAL E LOGARITMO DISCRETO

Uma das atividades essenciais da condição humana é sem dúvida a comunicação. Comunica-se para trocar ideias, expor nosso pensamento, fazer contratos e relatar fatos.

Para que uma comunicação seja eficiente alguns elementos são indispensáveis na comunicação: uma mensagem, um emissor para transmitir a mensagem, um receptor para recebe-la e um canal por onde a mensagem possa viajar partindo do emissor para o receptor. Além destes elementos o diálogo pode assumir várias características que dependem da intenção dos envolvidos. A comunicação pode ser pública, onde o receptor se constitui de um grande número de pessoas, ou privada, quando o emissor e o receptor constituem-se de duas pessoas e não desejam que mais ninguém conheça a mensagem. Uma das dificuldades na comunicação privada ocorre quando o canal disponível não é seguro, possibilitando o aparecimento de um interceptor quebrando o sigilo do diálogo. Quando é inviável obter um canal seguro, o emissor e o receptor podem combinar um código que possa embaralhar a mensagem antes de ser enviada, desta forma o interceptor não poderá entendê-la. Esta prática que consiste em codificar e decodificar mensagens é chamado de criptografia. Abaixo apresenta-se com mais detalhes o significado, aplicação e em que consiste a segurança de um método de criptografia.

2.1 CRIPTOGRAFIA SIMÉTRICA E ASSIMÉTRICA

A palavra criptografia é derivada de duas palavras gregas, *kryptós* que significa oculto ou secreto e *gráphein* que significa escrita. Criptografia é a palavra utilizada para descrever a ciência, ou “arte” como alguns preferem, de escrever textos em forma de códigos afim de tornar seu significado original oculto.

O exemplo mais simples de criptografia é trocar cada letra de um texto por um número, símbolo ou outra letra. Um computador, por exemplo, utiliza-se de uma codificação destas para trabalhar em sua matriz de cálculos, este entende cada carácter digitado no teclado como um número de 8 dígitos na representação binária. A tabela 2.1 apresenta o código ASCII (American Standard Code for Information Interchange, que em português significa "Código Padrão Americano para o Intercâmbio de Informação"), neste código os primeiros caracteres são utilizados como controle.

Embora seja fácil utilizar este tipo de codificação, ele é facilmente decifrado a partir da frequência com que cada símbolo aparece. Por exemplo, na língua portuguesa as vogais são as letras que mais aparecem em textos, de modo que um símbolo frequente em uma mensagem codificada dificilmente não será uma vogal, e não será diferente com os outros símbolos.

Além deste problema, os interlocutores precisam conhecer a tabela de símbolos para codificar e decodificar a mensagem que querem trocar. Para ser necessário o uso

Tabela 2.1: ASCII (American Standard Code for Information Interchange)

carac	dec	carac	dec	carac	dec	carac	dec	carac	dec	carac	dec	carac	dec
NUL	0	DC3	19	&	38	9	57	L	76	_	95	r	114
SOH	1	DC4	20	’	39	:	58	M	77	`	96	s	115
STX	2	NAK	21	(	40	;	59	N	78	a	97	t	116
ETX	3	SYN	22	)	41	<	60	O	79	b	98	u	117
EOT	4	ETB	23	*	42	=	61	P	80	c	99	v	118
ENQ	5	CAN	24	+	43	>	62	Q	81	d	100	w	119
ACK	6	EM	25	´	44	?	63	R	82	e	101	x	120
BEL	7	SUB	26	-	45	@	64	S	83	f	102	y	121
BS	8	ESC	27	.	46	A	65	T	84	g	103	z	122
TAB	9	FS	28	/	47	B	66	U	85	h	104	{	123
LF	10	GS	29	0	48	C	67	V	86	i	105	|	124
VT	11	RS	30	1	49	D	68	W	87	j	106	}	125
FF	12	US	31	2	50	E	69	X	88	k	107	~	126
CR	13	SPACE	32	3	51	F	70	Y	89	l	108	DEL	127
SO	14	!	33	4	52	G	71	Z	90	m	109		
SI	15	“	34	5	53	H	72	[	91	n	100		
DLE	16	#	35	6	54	I	73	\	92	o	111		
DC1	17	$	36	7	55	J	74	]	93	p	112		
DC2	18	%	37	8	56	K	75	^	94	q	113		

da criptografia supõem-se que o canal por onde a mensagem viaja não é seguro, de modo que a tabela dos símbolos não pode ser enviada por ele. Como exemplo, a criptografia é utilizada por lojas e bancos para compras e transações online, e normalmente os interlocutores não podem se encontrar pessoalmente para trocar informações sobre o algoritmo de criptografia que utilizarão.

Qualquer método de criptografia existente, pode ser classificado em um dentro dois grandes grupos: simétrico e assimétrico (ou de chave pública). Para explicar o que vem a ser estes grupos é necessário conhecer mais detalhadamente o que vem a ser um algoritmo de criptografia.

Definição 2.1. *Seja M o conjunto contendo a mensagem a ser codificada, C o conjunto que contem as mensagens codificadas, K_c e K_d conjuntos cujos elementos são denominados chaves. Um algoritmo de criptografia consiste de uma função $e : M \times K_c \to C$ e uma função $d : C \times K_d \to M$ onde, para qualquer $k_c \in K_c$ existe $k_d \in K_d$ tal que para todo $m \in M$, $d(e(m, k_c), k_d) = m$*

Na prática, para que um algoritmo de criptografia com os parâmetros (M, C, K_c, K_d, e, d) como na definição acima seja viável é preciso que satisfaça as seguintes condições:

1. Para qualquer chave $k_c \in K_c$ e qualquer texto $m \in M$, deve ser relativamente fácil calcular $e(m, k)$;

2. Para qualquer chave $k_d \in K_d$ e qualquer texto codificado $c \in C$, deve ser relativamente fácil calcular o texto decodificado $d(k, c)$;

3. Dados um ou mais textos cifrados $c_1, c_2, \cdots, c_n \in C$ usando a chave k_c, deve ser inviável calcular os respectivos textos decodificados $d(k_d, c_1), d(k_d, c_2), \cdots, d(k_d, c_n)$ sem a chave k_d apropriada;

4. Dados um ou mais pares de textos com suas respectivas codificações $(m_1, c_1), (m_2, c_2), \cdots, (m_n, c_n)$ usando a chave k_c, deve ser inviável decodificar uma texto codificado c qualquer sem a chave apropriada.

Na definição acima os elementos k_c e k_d são denominados respectivamente de chave de codificação e chave de decodificação.

No exemplo anterior M pode ser entendido como o conjunto das letras do alfabeto, de palavras ou de qualquer sequência de letras, C o conjunto de símbolos ou sequência de símbolos conforme a escolha de M e $K_c = K_d$ o conjunto das tabelas, ou funções invertíveis que relaciona cada letra com um símbolo. Observe que nesta situação sempre teremos $k_c = k_d$.

Dizemos que um algoritmo de criptografia é simétrico quando é viável (em termos da teoria e tecnologia presente) descobrir a chave de decodificação a partir da chave de codificação, caso contrário dizemos que o método é assimétrico ou de chave pública.

A ideia da criptografia de chave pública é que o receptor pode usar um método como o descrito na definição 2.1 e tornar pública a chave k_c. O emissor que deseja enviar uma mensagem m ao receptor com segurança deve calcular $e(m, k_c)$ e o enviar ao receptor. O receptor, único que tem conhecimento de k_d, lê a mensagem calculando $d(e(m, k_c), k_d) = m$. Qualquer interceptor conseguirá ler apenas k_c e $e(m, k_c)$, mas não conseguiria obter k_d para ler a mensagem.

A questão de um método de criptografia ser simétrico ou assimétrico muito depende dos recursos tecnológicos e teorias presentes no momento. Hoje um método pode ser considerado assimétrico e amanhã alguém descobrir um algoritmo para decifrá-lo, o tornando simétrico. Também é possível que o algoritmo seja seguro em geral, mas a má escolha de seus parâmetros o torne inseguro. Alguns desse detalhes da segurança da criptografia em curvas elípticas é discutido durante o trabalho.

2.2 Algoritmo de Criptografia de ElGamal

Em 1985 Taher ElGamal publicou em [7] um algoritmo de criptografia de chave pública baseado na estrutura de grupo, mas tarde este mesmo algoritmo foi utilizado para codificar mensagens utilizando grupos de pontos de curvas elípticas. Para entender melhor este algoritmo é necessário conhecer algumas propriedades da estrutura de grupos.

Definição 2.2 (Grupo)**.** *Um grupo é um conjunto $G \neq \emptyset$ com uma composição*

$$\begin{array}{rcl} \circ \ : G \times G & \to & G \\ (a, b) & \mapsto & a \circ b \end{array}$$

com as seguintes propriedades:

1) Associatividade: se $a, b, c \in G$, então

$$a \circ (b \circ c) = (a \circ b) \circ c;$$

2) Identidade: Existe um único $e \in G$ tal que se $a \in G$, então

$$a \circ e = e \circ a = a;$$

3) Inverso: se $a \in G$, então existe um único $b \in G$ tal que

$$a \circ b = b \circ a = e,$$

o elemento b é chamado inverso de a e denotado por $b = a^{-1}$.

Denota-se o grupo por $(G, \circ)$ ou apenas G quando a composição $\circ$ é subentendida. Um grupo $(G, \circ)$ é dito abeliano (ou comutativo) se possui a seguinte propriedade.

4) Comutatividade: se $a, b \in G$, então

$$a \circ b = b \circ a.$$

No caso em que G é um grupo aditivo, isto é, quando a composição $\circ$ é a adição $+$, a identidade (aditiva) de $(G, +)$ é denotada por 0 e o inverso (aditivo) de um elementos $a \in G$ por $-a$.

Definição 2.3. *Se $(G, \circ)$ é um grupo, $a \in G$ e $n \in \mathbb{Z}$ defini-se a^n como:*

i) se $n \geq 0$, então $\left\{ \begin{array}{rcl} a^0 & = & e \\ a^{n+1} & = & a^n \circ a \end{array} \right.$

ii) se $n < 0$, então $a^n = (a^{-n})^{-1}$.

Proposição 2.4. *Se G é um grupo e $a \in G$, então para todo $m, n \in \mathbb{Z}$ tem-se:*

i) $a^{n+m} = a^n \circ a^m$;

ii) $a^{mn} = (a^m)^n$.

Quando não há risco de ambiguidade a composição $a \circ b$ é simplesmente denotada por ab. Quando G é aditivo com composição "+", o elemento $a^n \in G$ é denotado como na e proposição 2.4 fica reformulada como:

Proposição 2.5. *Se $(G, +)$ é um grupo aditivo e $a \in G$, então para todo $m, n \in \mathbb{Z}$ temos:*

i) $(m + n)a = ma + na$;

ii) $(mn)a = m(na)$.

Intuitivamente falando $na = \underbrace{a + a + \cdots + a}_{n-vezes}$.

Supondo que uma mensagem m precise ser enviada de um emissor para um receptor e que estes tenham em mãos um grupo G, pode-se usar o algoritmo de criptografia de ElGamal seguindo os seguintes passos:

1. Primeiramente supõem-se que m é um elemento de G, caso isto não ocorra pode-se fazer uma pré-codificação com um algoritmo de criptografia simétrico como no exemplo da mudança de carácter.

2. O emissor e o receptor devem escolher um elemento $g \in G$, eles podem fazer isto usando o canal que possuem para enviar a mensagem codificada.

3. Neste passo o receptor deve escolher um inteiro positivo a, calcular g^a e enviá-lo para o emissor.

4. De posse da mensagem m e do elemento g^a, o emissor deve escolher outro inteiro positivo b, calcular $c_1 = g^b$ e $c_2 = (g^a)^b m = g^{ab} m$ e enviar o par (c_1, c_2) para o receptor.

5. Por fim, o receptor deve calcular $c_1^{-a} c_2 = g^{-ab} g^{ab} m = m$.

Fazendo um paralelo deste método com a definição 2.1 tem-se $M = G$, $C = G \times G$, $K_C = G$, $K_D = \mathbb{Z}$, $e = e_b(m, g^a) = (g^b, (g^a)^b m)$ é a função de codificação, $d((c_1, c_2), a) = c_2 c_1^{-a}$ é a função de decodificação, g^a é a chave pública e a é a chave privada. A tabela 2.2 resume este algoritmo.

Dadas as recomendações feitas para o algoritmo de criptografia anteriormente, a operação em G e o calculo do inverso de um elemento devem ser fáceis. Outra questão é o cálculo de g^a onde $a \in \mathbb{Z}$ e $g \in G$. Na prática, os inteiros a e b devem ser suficientemente grandes para evitar a quebra de informações por terceiros. A grandeza destes inteiros está relacionada com a tecnologia atual e aos resultados conhecidos para a quebra do código de ElGamal. Neste trabalho os termos "relativamente grandes" e "relativamente pequenos" são usados para relacionar a dependência dos parâmetros da criptografia com o desenvolvimento tecnológico.

O elemento g^a pode ser calculado eficientemente se a é escrito na sua representação binária. O algoritmo 2.6 apresenta este cálculo.

Tabela 2.2: Algoritmo de Criptografia de ElGamal

Parâmetros Públicos	
Um grupo G com ordem relativamente grande e um elementos $g \in G$ com ordem N também relativamente grande	
Receptor	**Emissor**
Criação da chave privada	
Escolhe a chave privada $1 < a < N$ Calcula $A = g^a$ em G Torna público a chave A	
Codificação	
	Escolhe o texto $m \in G$ Escolhe aleatoriamente$1 < b < N$ Usa A para calcular $c_1 = g^b$ e $c_2 = mA^b$ Envia (c_1, c_2) ao receptor
Decodificação	
Calcula $c_2 \cdot (c_1^a)^{-1}$. Este é igual a m	

Algoritmo 2.6. *Potenciação de Elementos de Grupos*

Entrada*: Um elemento $g \in G$ de um grupo e um inteiro $n \in Z$*

Saída*: O elementos $h = g^n \in G$*

1. Coloque $h \leftarrow e$;

2. se $n < 0$ coloque $N \leftarrow -n$ e $z \leftarrow g^{-1}$, se não coloque $N \leftarrow n$ e $z \leftarrow g$;

3. se $N \equiv 1 \mod 2$, coloque $h \leftarrow h.z$;

4. coloque $N \leftarrow \lfloor N/2 \rfloor$;

5. se $N \neq 0$ coloque $z \leftarrow z \cdot z$ e volte ao passo 3.

O tempo necessário para que o algoritmo 2.6 seja executado dependerá do tamanho do inteiro n de sua entrada, assim este tempo é um função de n no conjunto dos números reais, a esta função é denominada *custo de execução do algoritmo* . A função custo de um algoritmo pode ser encontrada conhecendo o tempo médio necessário para se executar as operações com 1 ou 2 bits. Normalmente a função custo apresenta uma expressão complicada, não muito significativa para analisar o algoritmo, neste caso o que se procura é um custo *assintótico* de execução. O custo assintótico é uma função que possui um comportamento semelhante à função custo do algoritmo. Isto pode ser feito usando a seguinte definição.

Definição 2.7. *Sejam $f, g : \mathbb{Z} \to \mathbb{R}$, com $g(n)$ não-negativa para todo n. Dizemos que*

$f(n) = O(g(n))$ *quando existem constantes* C *e* c *tais que*

$$|f(n)| \leq Cg(n)$$

para todo $n \geq c$*.*

Considerando que o tempo médio necessário para fazer uma operação em G é $\mathcal{G}$ (em unidades de tempos como: segundos, minutos ou horas), e que a representação binária do inteiro n possui em média metade dos dígitos iguais a zero, o algoritmo 2.6 levará um tempo de $0,5\mathcal{G}\log_2 n$ operações em G no passo 3 e $\mathcal{G}\log_2 n$ no passo 5, assim o custo deste algoritmo é $O(1,5\mathcal{G}\log_2 n)$.

Um resultado fácil de verificar referente a definição 2.7 2.7 que simplifica esta expressão é:

Proposição 2.8. *Se* $f_1(n) = O(g_1(n))$ *e* $f_2(n) = O(g_2(n))$*, então*

$$(f_1 + f_2)(n) = O(\max\{g_1(n), g_2(n)\})$$

e

$$(f_1 f_2)(n) = O((g_1 g_2)(n)).$$

Nesta situação o custo assintótico de 2.6 é $O(\mathcal{G}\log_2 n)$.

Se a comunicação utilizando-se o algoritmo de ElGamal for grampeada o interceptor conseguira obter os parâmetros g, g^a, g^b e $g^{ab}m$, mas não conhecerá de imediato os valores de a, b, g^{ab} ou m. O interceptor pode tentar obter o elemento g^{-ab} a partir de g, g^a, g^b e calcular $g^{-ab}g^{ab}m = m$ para encontrar a mensagem, mas tratar este problema é em geral difícil, no sentido de seu tempo de execução ser alto. Uma outra forma de abordá-lo é tentar obter a a partir de g e g^a e calcular g^{ab} utilizando-se de g^b, este por sua vez é uma abordagem mais geral do problema e igualmente difícil. Tais dificuldades são melhor explicadas na próxima seção

Estes dois problemas apresentados no parágrafo anterior são chamados de problema de Diffie-Hellman (DHP) e problema do logaritmo discreto (DLP). Uma abordagem mais específica a estes problemas é dada na próxima seção.

2.3 Problema de Diffie-Hellman e Problema do Logaritmo Discreto

Definição 2.9 (DHP)**.** *Seja* G *um grupo,* $g \in G$ *e* $a, b \in \mathbb{Z}$*. O problema de Diffie-Hellman (DHP - Diffie-Hellman Problem) é o problema de encontrar* g^{ab} *conhecendo apenas* g^a *e* g^b*.*

Definição 2.10 (DLP)**.** *Seja* G *um grupo e* $g, h \in G$*. O problema do logaritmo discreto (DLP - Discrete Logarithm Problem) é o problema de encontrar o menor inteiro* x *em valor absoluto tal que*

$$g^x = h \tag{2.1}$$

Note que o DHP é exatamente o problema que apresentamos na seção anterior para decifrar o algoritmo de criptografia de ElGamal, mais que isso, em [8] encontra-se a demonstração de que o DHP e o problema de decifrar o código ElGamal são equivalentes, isto significa que saber resolver o DHP de modo fácil implica em quebrar o código de ElGamal e, se é possível quebrar este código pode-se resolver o DHP facilmente.

O problema do logaritmo discreto (DLP) é um pouco mais abrangente que o DHP. Suponha que sabemos resolver o DLP de modo fácil e temos $g, g^a, g^b \in G$ um grupo, neste caso pode-se tentar resolver a equação $g^x = g^a$ e calcular $(g^b)^x = (g^b)^a = g^{ab}$. Isto significa que resolver o DLP, implicando em resolver o DHP e portanto decifrar o sistema de criptografia de ElGamal. Entretanto a implicação contrária é um problema em aberto, ou seja, não sabemos que o DLP e o problema de decifrar o código de ElGamal são equivalentes.

O DLP leva este nome por sua dificuldade se encontrar em grupos discretos. Por exemplo, tomando $(\mathbb{R}^*, \cdot)$ e $g, h \in \mathbb{R}^*$, $g > 0$, $g \neq 1$ não há grandes dificuldade em resolver a equação $g^x = h$.

Outras questão pertinente é a possibilidade da equação $g^x = h$ não ter solução, isto ocorre quando h não pertence ao subgrupo de G gerado por g. Outro ponto importante, é que se a ordem de g é n e x_0 é uma solução de $g^x = h$, então qualquer inteiros da forma $x_0 + tn$ com $t \in \mathbb{Z}$ é uma solução para a mesma equação. Em termos práticos, basta apenas encontrar, dentre todas as soluções de $g^x = h$ quando existem, a menor delas, desta forma diz-se que o logaritmo discreto de h por g é o menor inteiro x em valor absoluto que satisfaz $g^x = h$.

Abordando o DHP resolvendo um DLP pode-se garantir que o problema sempre terá solução, desta forma g será considerado um gerador de G nas linhas subsequentes.

A primeira maneira de abordar o DLP na equação $g^x = h$ é calcular a lista $g, g^2, g^3, \cdots$ e comparar cada elemento desta com h. O cálculo de g^n possui o custo de $O(\mathcal{G} \log_2 n)$ enquanto que o custo para calcular esta lista até g^n é $O(\mathcal{G}n)$. Supondo n da ordem de 10^{150} e $\mathcal{G} = 10^{-3}$ segundos, tem-se que $\mathcal{G} \log_2 n$ equivale a menos de um segundo enquanto que $\mathcal{G}n$ ultrapassa $3,17 \times 10^{139}$ anos.

O valor de $\log_2 n$ é aproximadamente a quantidade de dígitos da representação binária de n, e por consequência o tamanho da entrada do algoritmo 2.6, assim a expressão $\mathcal{G} \log_2 n$ é uma função polinomial da entrada do algoritmo. Algoritmos cujo custo sejam uma função polinomial são normalmente viáveis de executar, diz-se neste caso que o custo do algoritmo é *polinomial*. Note agora que $\mathcal{G}n = \mathcal{G}2^{\log_2 n}$ é uma função exponencial da entrada. Algoritmos cujo custo é uma função exponencial da entrada são normalmente inviáveis de se executar, podendo levar anos ou milênios para darem uma resposta, diz-se nesta situação que o custo do algoritmo é *exponencial*.

Entre o custo polinomial e exponencial há ainda o que denomina-se custo *subexponencial*. Este por sua vez não é tão lento como o custo exponencial, porem seu tempo de execução ultrapassa qualquer algoritmo cujo custo seja polinomial. O próximo teorema apresenta uma forma de abordar o problema do logaritmo discreto a um custo subexponencial.

Teorema 2.11 (Shank's Babystep-Giantstep Algorithm)**.** *Seja G um grupo e seja $g \in G$ um elemento de ordem $N \geq 2$. O algoritmo que segue resolve o problema do logaritmo discreto em $O(\sqrt{N}\log N)$ passos.*

1. *Seja $n = \lfloor 1 + \sqrt{N} \rfloor$,em particular $n > \sqrt{N}$;*
2. *Calcule as duas listas abaixo:*

$$\begin{aligned} &\textit{Lista 1:} \quad e, g, g^2, \cdots, g^n \\ &\textit{Lista 2:} \quad h, hg^{-n}, hg^{-2n}, \cdots hg^{-n^2}; \end{aligned}$$

3. *Encontre dois elementos, um em cada lista, tais que*

$$g^i = hg^{-jn}; \tag{2.2}$$

4. *$x = i + jn$ é a solução de $g^x = h$.*

Demonstração: Se i e j são inteiros que satisfazem (2.2) então fazendo a composição de g^{jn} em ambos os lados desta igualdade tem-se $g^{i+jn} = h$. A demonstração da existência do par i, j, da unicidade de $i + jn$ bem como o custo $O(\sqrt{N}\log N)$ para executar este algoritmo pode ser encontrada em [16]. □

Conhecendo a fatoração da ordem de g pode-se resolver o DLP mais eficientemente usando o Babystep-Giantestep para cada fator primo e juntando estas informações com o teorema chinês do resto [9]. Este algoritmo é apresentado mais detalhadamente no próximo teorema.

Teorema 2.12 (Pohlig-Hellman Algorithm)**.** *Seja G um grupo e suponha conhecido um algoritmo para resolver o problema do logaritmo discreto em G para qualquer elemento cuja ordem seja potência de um número primo. Mais exatamente, se $g \in G$ possui ordem q^e com q primo, suponha que seja possível resolver $g^x = h$ em $O(S_{q^e})$ passos.*

Seja agora $g \in G$ um elemento de ordem N, e suponha que a fatoração de N seja

$$N = q_1^{e_1} q_2^{e_2} \cdots q_t^{e_t}.$$

Então o DLP $g^x = h$ pode ser resolvido em

$$O\left(\sum_{i=1}^{t} S_{q_i^{e_i}} + \log N\right)$$

passo usando o seguinte procedimento.

1. *Para cada $1 \leq i \leq t$ seja*

$$g_i = g^{N/q_i^{e_i}} \text{ e } h_i = h^{N/q_i^{e_i}}.$$

Note que assim a ordem de g_i é $q_i^{e_i}$, deste modo é possível usar o algoritmo dado para resolver

$$g_i^y = h_i, \tag{2.3}$$

seja $y = y_i$ a solução de (2.3);

2. *Use o teorema chinês do resto para resolver*

$$x \equiv y_i(\mod q_i^{e_i})$$

com $i = 1, 2, \cdots, t$.

Demonstração: Ver [8], teorema 2.32. □

Pode-se reescrever estes mesmos resultados no caso em que o grupo G é aditivo sem alterar os argumentos.

É importante observar neste ponto que os algoritmos apresentados anteriormente para resolver o DLP são gerais para qualquer grupo G de ordem finita que tomarmos, isto não significa que resolver o DLP está restrito a estes algoritmos em qualquer grupo. Por exemplo, se $G = \mathbb{Z}_p$ com p primo, então a solução de $x \cdot g = h$ é $g_1 \cdot h$ onde g_1 é o menor inteiro positivo tal que $g \cdot g_1 \equiv 1 \mod p$. Além disso, g_1 pode ser encontrado facilmente utilizando o algoritmo estendido de Euclides [3], algoritmo 1.3.1, a um custo de $O(\log_2^3 p)$.

Visto tudo isso pode-se estabelecer alguns critérios que devem ser satisfeitos para utilizar o método de criptografia de ElGamal de modo seguro:

1. Resolver o DLP no grupo G escolhido deve ser algo difícil, porem efetuar as operações neste conjunto não pode se tornar um impedimento para a comunicação.

2. A ordem do elemento $g \in G$ deve ser suficientemente grande para que a lista $g, g^2, g^3, \cdots$ não seja facilmente calculada. É uma consequência imediata disto que a ordem de G seja igualmente grande.

3. A ordem de g não pode ser facilmente fatorada em pequenos primos.

4. Os inteiros a e b escolhidos pelos interlocutores também não devem ser pequenos.

3 CORPOS

3.1 ANÉIS E CORPOS

Definição 3.1 (Anel)**.** *Um anel é um conjunto* $\mathbf{R} \neq \emptyset$ *com duas operações:*
Adição:

$$\begin{aligned} + \ : \mathbf{R} \times \mathbf{R} &\to \mathbf{R} \\ (a, b) &\mapsto a + b \end{aligned}$$

e
Multiplicação:

$$\begin{aligned} \cdot \ : \mathbf{R} \times \mathbf{R} &\to \mathbf{R} \\ (a, b) &\mapsto ab \end{aligned}$$

com as seguintes propriedades:

A) $(\mathbf{R}, +)$ *é um grupo comutativo;*

M) Associatividade da Multiplicação: se $a, b, c \in \mathbf{R}$*, então*

$$a(bc) = (ab)c;$$

D) Distributividade: se $a, b, c \in \mathbf{R}$*, então*

$$a(b + c) = ab + ac;$$

$$(a + b)c = ac + bc.$$

O anel definido acima é denotado por $(\mathbf{R}, +, \cdot)$ ou simplesmente por $\mathbf{R}$ quando as operações são subentendidas. É fácil verificar que:

Proposição 3.2. *Se* $\mathbf{R}$ *é um anel e* $a, b \in \mathbf{R}$*, então:*

i) $a0 = 0a = 0$*;*

ii) $a(-b) = (-a)b = -ab$*.*

Definição 3.3. *Seja* $\mathbf{R} = (\mathbf{R}, +, \cdot)$ *um anel. Dize-se que* $\mathbf{R}$ *é um:*

1. **Anel com identidade:** *se existe* $1 \in \mathbf{R}$*,* $1 \neq 0$ *tal que para todo* $a \in \mathbf{R}$*,* $1a = a1 = a$*;*
2. **Anel Comutativo:** *se* $a, b \in \mathbf{R}$*, então* $ab = ba$*;*
3. **Anel sem divisor de zero:** *se* $a, b \in \mathbf{R}$ *e* $ab = 0$*, então* $a = 0$ *ou* $b = 0$*. Caso contrário dize-se que* $\mathbf{R}$ *é um* **anel com divisor de zero***;*

4. **Anel de divisão:** *se* $(\mathbf{R}^*, \cdot)$ *é um grupo, onde* $\mathbf{R}^* = \mathbf{R} - \{0\}$*;*

5. **Corpo:** *se* $(\mathbf{R}^*, \cdot)$ *é um grupo comutativo;*

6. **Domínio de Integridade:** *se* $\mathbf{R}$ *é comutativo, com identidade e sem divisor de zero.*

Definição 3.4 (Subanel)**.** *Um subconjunto* $\mathbf{S} \neq \emptyset$ *de um anel* $(\mathbf{R}, +, \cdot)$ *é dito ser um subanel quando satisfaz:*

i) se $a, b \in \mathbf{S}$*, então* $a + b \in \mathbf{S}$*;*

ii) se $a, b \in \mathbf{S}$*, então* $ab \in \mathbf{S}$*;*

iii) as operações "$+$*" e "*$\cdot$*" restritas por (i) e (ii) em* $\mathbf{S}$ *faz deste um anel.*

Nesta situação $(\mathbf{R}, +, \cdot)$ é dito uma extensão de $(\mathbf{S}, +, \cdot)$. Não é difícil ver que a identidade aditiva e o inverso aditivo de um elemento $a \in \mathbf{S}$ são os mesmo tanto em $\mathbf{R}$ quanto em $\mathbf{S}$.

Se $\mathbf{K}$ é corpo, nem sempre um subanel de $\mathbf{K}$ também é um corpo, mas se $\mathbf{F}$ é um subanel de $\mathbf{K}$ que é corpo diz-se que $\mathbf{F}$ é um subcorpo de $\mathbf{K}$ e $\mathbf{K}$ uma extensão de corpos de $\mathbf{F}$, neste caso diz-se que $\mathbf{K}|\mathbf{F}$ é uma extensão de corpos.

Exemplo 1. O conjunto dos números inteiros denotado por $\mathbb{Z} = \{0, \pm 1, \pm 2, \cdots\}$ é um exemplo de anel que é domínio de integridade e não é um corpo. Os conjuntos dos números racionais $\mathbb{Q}$, reais $\mathbb{R}$ e complexos $\mathbb{C}$ são exemplos de anéis que são corpos. Observamos que $\mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$ é uma sequências de subanéis, além disso temos as extensões de corpos $\mathbb{R}|\mathbb{Q}$ e $\mathbb{C}|\mathbb{R}$.

Exemplo 2 (Corpo Quadrático)**.** Seja $d \in \mathbb{Z}$ tal que se $d \neq n^2$ para todo inteiro $n > 1$, seja $\mathbb{Z}[\sqrt{d}] = \{a + b\sqrt{q} : ab \in \mathbb{Z}\}$. Dadas as operações

$$\begin{array}{rrcl} + : & \mathbb{Z}[\sqrt{d}] \times \mathbb{Z}[\sqrt{d}] & \to & \mathbb{Z}[\sqrt{d}] \\ & (a_1 + b_1\sqrt{d}, a_2 + b_2\sqrt{d}) & \mapsto & (a_1 + a_2) + (b_1 + b_2)\sqrt{d} \end{array}$$

e

$$\begin{array}{rrcl} \cdot : & \mathbb{Z}[\sqrt{d}] \times \mathbb{Z}[\sqrt{d}] & \to & \mathbb{Z}[\sqrt{d}] \\ & (a_1 + b_1\sqrt{d}, a_2 + b_2\sqrt{d}) & \mapsto & (a_1a_2 + b_1b_2d) + (a_1b_2 + a_2b_1)\sqrt{d}, \end{array}$$

$(\mathbb{Z}[\sqrt{d}], +, \cdot)$ é um domínio de integridade. Considerando as mesmas operações em $\mathbb{Q}(\sqrt{d}) = \{a + b\sqrt{d} : a, b \in \mathbb{Q}\}$, mas com $a_1, a_2, b_1, b_2 \in \mathbb{Q}$ e d não-quadrado em $\mathbb{Q}$, $(\mathbb{Q}(\sqrt{d}), +, \cdot)$ é ainda um corpo denominado *corpo quadrático* de d. Em particular, se $d < 0$, $\mathbb{Q}(\sqrt{d})$ também é chamado de corpo quadrático imaginário, ou complexo.

Exemplo 3. Seja $\mathbf{R} = \mathbb{Q}(i, j, k)$, o conjunto das expressões da forma $a + bi + cj + dk$, com $a, b, c, d \in \mathbb{Q}$. $\mathbf{R}$ é um anel de divisão com as operações
Adição

$$\begin{array}{rrcl} + : & \mathbf{R} \times \mathbf{R} & \to & \mathbf{R} \\ & (x, y) & \mapsto & x + y \end{array}$$

e
Multiplicação

$$\begin{array}{rccc} \cdot : & \mathbf{R}\times\mathbf{R} & \to & \mathbf{R} \\ & (x,y) & \mapsto & xy \end{array}$$

onde se $x = a + bi + cj + dk$ e $y = a' + b'i + c'j + d'k$, então

$$x + y = (a + a') + (b + b')i + (c + c')j + (d + d')k$$

e

$$xy = \alpha + \beta i + \gamma j + \delta k.$$

onde

$$\begin{array}{rcl} \alpha & = & aa' + bb' + cc' + dd' \\ \beta & = & ab' + ba' + cd' - dc' \\ \gamma & = & ac' - bd' + ca' + db' \\ \delta & = & ad' + bc' - cb' + da' \end{array}$$

Tem-se ainda que $i^2 = j^2 = k^2 = ijk = -1$, $\overline{x} = a - bi - cj - dk$ e $x\overline{x} = a^2 + b^2 + c^2 + d^2$.

Definição 3.5 (Homomorfismo de anéis)**.** *Um* **homomorfismo** *do anel* $\mathbf{R} = (\mathbf{R}, +, \cdot)$ *para o anel* $\mathbf{S} = (\mathbf{S}, +', \cdot')$ *é uma função*

$$f : \mathbf{R} \to \mathbf{S}$$

tal que

i) $f(a + b) = f(a) + f(b)$*;*

ii) $f(ab) = f(a)f(b)$.

Se f *é bijetora diz-se que* f *é um* **isomorfismo** *de anéis. Dois anéis* $\mathbf{R}$ *e* $\mathbf{S}$ *são ditos ser* **isomorfos***, denotado por* $\mathbf{R} \cong \mathbf{S}$*, se existe um isomorfismo entre eles.*

As vezes exigi-se ainda que $f(1) = 1'$ quando $\mathbf{R}$ e $\mathbf{S}$ possui identidades respectivamente 1 e $1'$.

É fácil verificar que:

Proposição 3.6. *Se* $\mathbf{R}$ *e* $\mathbf{S}$ *são anéis e* $f : \mathbf{R} \to \mathbf{S}$ *é um homomorfismo de* $\mathbf{R}$ *em* $\mathbf{S}$*, então:*

i) $f(0) = 0$*;*

ii) $f(-a) = -f(a)$*;*

ii) Se $\mathbf{R}$ *e* $\mathbf{S}$ *são com identidade e* $\mathbf{S}$ *é sem divisor de zero, então* $f(1) = 1$.

Definição 3.7 (Ideal)**.** *Um subconjunto* $\mathbf{I} \neq \emptyset$ *de um anel* $\mathbf{R}$ *é um ideal se satisfaz:*

1. *Se* $a, b \in \mathbf{I}$*, então* $a - b \in \mathbf{I}$*;*

2. *Se* $a \in \mathbf{R}$ *e* $b \in \mathbf{I}$*, então* $ab, ba \in \mathbf{I}$*.*

É óbvio que se $\mathbf{I}$ é um ideal de um anel $\mathbf{R}$, então $\mathbf{I}$ é um subanel de $\mathbf{R}$. Além disso $\{0\}$ e $\mathbf{R}$ são ideais do anel $\mathbf{R}$. Por outro lado, se $\mathbf{F}$ é um corpo, $\mathbf{I}$ é um ideal de $\mathbf{F}$ e existe $a \in \mathbf{I}$ tal que $a \neq 0$, então para qualquer $x \in \mathbf{F}$, $x = x(a^{-1}a) \in \mathbf{I}$ o que implica $\mathbf{I} = \mathbf{F}$.

Se $\mathbf{R}$ é um anel, um ideal $\mathbf{I}$ de $\mathbf{R}$ diferente de $\mathbf{R}$ e $\{0\}$ é denominado um ideal não-trivial de $\mathbf{R}$. Isto significa que corpos não possuem ideais não-triviais.

Por exemplo $\mathbb{Q}$ é subanel de $\mathbb{R}$ mas não é um ideal pois $\mathbb{R}$ é um corpo e $\mathbb{Q} \neq \mathbb{R}$ e $\mathbb{Q} \neq \{0\}$.

Se S é um subconjunto de $\mathbf{R}$, o menor ideal $\mathbf{I}$ de $\mathbf{R}$ contendo S é dito ser o ideal gerado por S, tal ideal existe e é único, pois $\mathbf{R}$ é um ideal contendo S e como a interseção de uma família não-vazia de ideais é um ideal, este ideal referido é nada mais do que a interseção da família dos ideais contendo S.

Quando S é formado por um único elemento a, $\mathbf{I}$ é dito ser um ideal principal e denotamos $\mathbf{I} = (a)$. Se $\mathbf{R}$ é um anel comutativo com identidade e S um subconjunto de $\mathbf{R}$ não-vazio, então $\mathbf{I} = \{a_1\alpha_1 + a_2\alpha_2 + \cdots + a_r\alpha_r : \alpha_i \in \mathbf{R}, a_i \in S\}$. Caso $S = \emptyset$, então $\mathbf{I} = \{0\}$.

Definição 3.8 (Anel Quociente)**.** *Seja* $\mathbf{R}$ *um anel e* $\mathbf{I}$ *um ideal de* $\mathbf{R}$*. Um elemento* $a \in \mathbf{R}$ *é congruente a um elemento* $b \in \mathbf{R}$ *módulo* $\mathbf{I}$ *se, e somente se,* $a - b \in \mathbf{I}$*, e denota-se* $a \equiv b \mod \mathbf{I}$*. Esta é uma relação de equivalência onde a classe de equivalência de um elemento* $a \in \mathbf{R}$ *é:*

$$\mathbf{cl}(a) = \{a + \alpha : \alpha \in \mathbf{I}\}.$$

Dada a relação acima, o conjunto quociente de $\mathbf{R}$ *por* $\mathbf{I}$ *é definido como*

$$\mathbf{R}/\mathbf{I} = \{\mathbf{cl}(a) : a \in \mathbf{R}\}.$$

Este conjunto é um anel, denominado **anel quociente** *de* $\mathbf{R}$ *por* $\mathbf{I}$*, com as operações bem definidas por*

$$\begin{array}{rccc} + & : \mathbf{R}/\mathbf{I} \times \mathbf{R}/\mathbf{I} & \to & \mathbf{R}/\mathbf{I} \\ & (\mathbf{cl}(a), \mathbf{cl}(b)) & \mapsto & \mathbf{cl}(a+b) \end{array}$$

e

$$\begin{array}{rccc} \cdot & : \mathbf{R}/\mathbf{I} \times \mathbf{R}/\mathbf{I} & \to & \mathbf{R}/\mathbf{I} \\ & (\mathbf{cl}(a), \mathbf{cl}(b)) & \mapsto & \mathbf{cl}(ab) \end{array}$$

A identidade aditiva de $\mathbf{R}/\mathbf{I}$ *é* $\mathbf{cl}(0) = \mathbf{I}$*. Além disso,* $-\mathbf{cl}(a) = \mathbf{cl}(-a)$ *para todo* $a \in \mathbf{R}$*. Se* $1 \in \mathbf{R}$ *e* $1 \notin \mathbf{I}$*, então* $\mathbf{R}/\mathbf{I}$ *é um anel com identidade* $1' = \mathbf{cl}(1)$*.*

Em $\mathbb{Z}$ todo ideal é principal e da forma $\mathbf{I} = n\mathbb{Z} = \{n.a : a \in \mathbb{Z}\}$ para algum $n \in \mathbb{Z}$ dependendo de $\mathbf{I}$. Um domínio de integridade onde todo ideal é principal é dito **domínio de ideais principais** (DIP), (ver [9] na página 86).

Dado $n \in \mathbb{Z}$, denota-se o anel quociente $\mathbb{Z}/n\mathbb{Z}$ por $\mathbb{Z}_n$. Pelo algoritmo da divisão de Euclides, para qualquer $a \in \mathbb{Z}$, existem únicos $q, r \in \mathbb{Z}$ tais que $a = nq + r$ e $0 \leq r < |n|$, ou ainda $a - r \in n\mathbb{Z}$, ou seja, qualquer elemento de $\mathbb{Z}_n$ pode ser representado por um inteiro não-negativo menor que n. Pode-se identificar $\mathbb{Z}_n$ por $\mathbb{Z}_n = \{0, 1, 2, \cdots, n-1\}$ com n elementos, de modo que facilite as operações neste conjunto.

Definição 3.9. *Um ideal* $\mathbf{I}$ *de um anel* $\mathbf{R}$ *é dito* **maximal** *se* $\mathbf{I} \neq \mathbf{R}$ *e sempre que* $\mathbf{I} \subseteq \mathbf{J} \subseteq \mathbf{R}$ *para qualquer ideal* $\mathbf{J}$ *de* $\mathbf{R}$*,* $\mathbf{J} = \mathbf{I}$ *ou* $\mathbf{J} = \mathbf{R}$*.*

Teorema 3.10. *Se* $\mathbf{R}$ *é um anel comutativo com identidade e* $\mathbf{I}$ *um ideal de* $\mathbf{R}$*, então* $\mathbf{R}/\mathbf{I}$ *é um corpo se, e somente se,* $\mathbf{I}$ *é maximal.*

Demonstração: Ver [14], teorema 1.3.8. □

Em $\mathbb{Z}$ um ideal $p\mathbb{Z}$ é maximal se, e somente se, p é primo, neste caso o anel quociente $\mathbb{Z}_p$ é um corpo. Neste corpo $p1 = 0$, isto defini a característica de um corpo que é apresentada a seguir.

Definição 3.11. *Se* $\mathbf{F}$ *é um corpo definimos a característica de* $\mathbf{F}$*, denotado por* $\mathbf{car}(\mathbf{F})$*, em dois casos:*

- **Caso 1:** $\mathbf{car}(\mathbf{F}) = 0$ *se não existe um inteiro positivo* n *tal que* $n \cdot 1 = 0$*;*
- **Caso 2:** $\mathbf{car}(\mathbf{F}) = n$ *se* n *é o menor inteiro positivo tal que* $n \cdot 1 = 0$*.*

Proposição 3.12. *Se* $\mathbf{F}$ *um corpo e* $\mathbf{car}(\mathbf{F}) = p \neq 0$*, então* p *é primo.*

Demonstração: Se existem inteiros positivos n_1 e n_2 tais que $\mathbf{car}(\mathbf{F}) = p = n_1 n_2$, então $0 = p1 = (n_1 n_2)1 = (n_1 1)(n_2 1)$, seque daí que $n_1 1 = 0$ ou $n_2 1 = 0$. Sendo $n_1, n_2 \leq p$ segue da definição de característica que $p = n_1$ ou $p = n_2$, logo p é primo □.

3.2 Extensão de Corpos

Das várias formas de se obter uma extensão de um corpo base, neste trabalho é usado apenas extensões obtidas por polinômios irredutíveis sobre um corpo. Desta forma tem-se um meio prático de representar e operar com os elementos destas extensões a partir dos elementos do corpo base.

Definição 3.13. *Seja* $\mathbf{R}$ *um anel e* $a_0, a_1, \cdots, a_n, \cdots$ *uma sequência de elementos de* $\mathbf{R}$ *com um número finito de elementos não-nulos. Um polinômio* f *em uma indeterminada* X *com coeficientes* $a_0, a_1, \cdots, a_n, \cdots$ *é uma expressão da forma*

$$f(X) = a_0 + a_1 X + a_2 X^2 + \cdots + a_n X^n + \cdots$$

Além disso, se n é o maior inteiro tal que $a_n \neq 0$, dizemos que n é o grau de f, e denota-se por $\mathbf{gr}(f) = n$. O grau do polinômio em que todos seus coeficientes são nulos não é definido.

O conjunto dos polinômios com coeficientes em um anel $\mathbf{R}$ em uma indeterminada X é denotado por $\mathbf{R}[X]$.

Dados dois polinômios

$$f(X) = a_0 + a_1X + a_2X^2 + \cdots + a_nX^n + \cdots,$$

$$g(X) = b_0 + b_1X + b_2X^2 + \cdots + b_nX^n + \cdots$$

em $\mathbf{R}[X]$, defini-se as operações de adição e multiplicação de f e g respectivamente por

$$(f+g)(X) = (a_0 + b_0) + (a_1 + b_1)X + (a_2 + b_2)X^2 + \cdots + (a_n + b_n)X^n \cdots$$

e

$$(fg)(X) = c_0 + c_1X + c_2X^2 + \cdots c_nX^n + \cdots$$

onde $c_n = a_0b_n + a_1b_{n-1} + a_2b_{n-2} + \cdots + a_nb_0 = \sum_{i+j=n} a_ib_j$.

De fato, $(\mathbf{R}[X], +, \cdot)$ é um anel, denominado *anel de polinômios* sobre $\mathbf{R}$.

Teorema 3.14. *Sobre anéis de polinômios temos:*

1. *Se $\mathbf{R}$ é um domínio de integridade, então $\mathbf{R}[X]$ é um domínio de integridade;*
2. *Se $\mathbf{F}$ é um corpo, então $\mathbf{F}[X]$ é um domínio de ideais principais.*

Demonstração: Ver [9], teorema 1.2. □

Observe que se $\mathbf{R}$ é um domínio de integridade, $f, g \in \mathbf{R}[X]$, $\mathbf{gr}(f) = n$ e $\mathbf{gr}(g) = m$, então $n + m$ é o maior inteiro tal que $c_{n+m} = a_nb_m \neq 0$, ou seja $\mathbf{gr}(fg) = n + m$.

Definição 3.15. *Se $\mathbf{R}$ um domínio de integridade e $a, b \in \mathbf{R}$, diz-se que a divide b, denotado por $a|b$, se existe $c \in \mathbf{R}$ tal que $b = ac$.*

Proposição 3.16. *Se $\mathbf{R}$ um domínio de integridade e $f, g \in \mathbf{R}[X]$ com $f, g \neq 0$, então:*

i) $\mathbf{gr}(fg) = \mathbf{gr}(f)\mathbf{gr}(g)$;

ii) $f|g \Rightarrow \mathbf{gr}(f) \leq \mathbf{gr}(g)$

Teorema 3.17. *Se $\mathbf{F}$ um corpo e $f, g \in \mathbf{F}[X]$ com $g \neq 0$, então existe e são únicos polinômios $q, r \in \mathbf{F}[X]$ com $\mathbf{gr}(r) < \mathbf{gr}(g)$ ou $r = 0$ tais que*

$$f = gq + r. \tag{3.1}$$

Demonstração: Ver [9], teorema 5.2. □

Definição 3.18. *Seja* $\mathbf{F}$ *um corpo e* $f \in \mathbf{F}[X]$. *Dizemos que* f *é um polinômio irredutível sobre* $\mathbf{F}$ *se:*

i) $\mathbf{gr}(f) \neq 0$ *e;*

ii) se $f = pq$ *tal que* $p, q \in \mathbf{F}[X]$, *então* $\mathbf{gr}(p) = 0$ *ou* $\mathbf{gr}(q) = 0$.

Teorema 3.19. *Seja* $\mathbf{F}$ *um corpo e* $\mathbf{I}$ *um ideal de* $\mathbf{F}[X]$ *gerado por* $f \in \mathbf{F}[X]$. $\mathbf{I}$ *é um ideal maximal se, e somente se,* f *é irredutível sobre* $\mathbf{F}[X]$.

Demonstração: Ver [9]. □

Pelos teoremas 3.10 e 3.19, se f é um polinômio irredutível sobre $\mathbf{F}$, então o anel quociente $\mathbf{F}[X]/(f)$ é um corpo. Além disso

$$\begin{array}{rcl} i \ : \mathbf{F} & \to & \mathbf{F}[X]/(f) \\ a & \mapsto & \mathbf{cl}(a), \end{array}$$

é um homomorfismo injetor, isto é, $\mathbf{F}[X]/(f)$ é uma extensão de $i(\mathbf{F})$, ou simplesmente de $\mathbf{F}$.

Se $\mathbf{K}$ é uma extensão de um corpo $\mathbf{F}$, então $\mathbf{K}$ é um espaço vetorial sobre $\mathbf{F}$. A dimensão deste espaço vetorial é denotado por $[\mathbf{K} : \mathbf{F}]$ e é chamada de grau de extensão de $\mathbf{K}$ sobre $\mathbf{F}$. Se $\mathbf{F}$ é um corpo e $f \in \mathbf{F}[X]$ um polinômio irredutível de grau m, então pelo teorema 3.17

$$\frac{\mathbf{F}[X]}{(f)} = \{a_0 + a_1X + \cdots + a_{m-1}X^{m-1} : a_0, a_1, \cdots, a_{m-1} \in \mathbf{F}\}.$$

Assim o conjunto $\{1, X, X^2, \cdots, X^{m-1}\}$ é uma base para o espaço vetorial $\mathbf{F}[X]/(f)$ sobre $\mathbf{F}$ e $[\mathbf{F}[X]/(f) : \mathbf{F}] = m$.

3.3 CORPOS FINITOS

Seja p um número primo e $f \in \mathbb{Z}_p[X]$ um polinômio irredutível de grau $m \geq 1$, então $\mathbf{K} = \dfrac{\mathbb{Z}_p[X]}{(f)}$ é um corpo. Pelo teorema 3.17, $\dfrac{\mathbb{Z}_p[X]}{(f)}$ possui p^m elementos, com

$$\frac{\mathbb{Z}_p[X]}{(f)} = \{g = a_0 + a_1X + \cdots + a_{m-1}X^{m-1} : a_0, a_1, \cdots, a_{m-1} \in \mathbb{Z}_p\}$$

Se $\mathbf{F}$ é um corpo finito com $\mathbf{car}(\mathbf{F}) = p$, tem homomorfismo injetor

$$\begin{array}{rcl} i : \ \mathbb{Z}_p & \to & \mathbf{F} \\ a & \mapsto & a1. \end{array}$$

Assim $\mathbb{Z}_p = i(\mathbb{Z}_p)$ é um subcorpo de $\mathbf{F}$. Disto segue:

Teorema 3.20. *Se* $\mathbf{F}$ *é um corpo finito com* q *elementos, então* $q = p^m$ *onde* p *é um número primo e* $m \geq 1$.

Demonstração: Como $\mathbf{F}$ é finito, $\mathbf{F}$ possui característica não nula p e portanto, a menos de isomorfismo, $\mathbb{Z}_p$ é um subcorpo de $\mathbf{F}$. Assim $\mathbf{F}$ é um espaço vetorial sobre $\mathbb{Z}_p$ com $m = [\mathbf{F} : \mathbb{Z}_p]$, desta forma $\mathbf{F}$ tem p^m elementos.

□

Este teorema diz ainda que o corpo com p elementos, onde p é primo, é único. É verdade ainda a unicidade de corpos finitos descrita no teorema abaixo.

Teorema 3.21. *Para qualquer número primo* p *e qualquer inteiro positivo* m*, existe, e é único (a menos de isomorfismo), o corpo finito com* p^m *elementos.*

Demonstração: Ver [1], página 407.

□

Pela unicidade do corpo finito com q elementos, este é simplesmente denotado por $\mathbb{F}_q$. Note que se f é um polinômio irredutível sobre $\mathbb{F}_q$ de grau k, então

$$\mathbb{F}_{q^k} = \frac{\mathbb{F}_q[X]}{(f)},$$

isto significa que o corpo finito $\mathbb{F}_{q^k}$, com q^k elementos, é uma extensão do corpo $\mathbb{F}_q$ de grau k. Neste trabalho a menção do corpo $\mathbb{F}_{q^k}$ será sempre entendida como a extensão do corpo $\mathbb{F}_q$ de grau k.

4 CURVAS ELÍPTICAS

Definição 4.1 (Curva Elíptica)**.** *Seja* $\mathbf{F}$ *um corpo e*

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \tag{4.1}$$

a equação de Weierstrass (na forma longa) com $a_1, a_3, a_2, a_4, a_6 \in \mathbf{F}$*. Se* $\mathbf{K}$ *é uma extensão de* $\mathbf{F}$*, a curva elíptica* E *sobre* $\mathbf{K}$ *com coeficientes em* $\mathbf{F}$*, denotada por* $E_{\mathbf{F}}(\mathbf{K})$*, é o conjunto*

$$E_{\mathbf{F}}(\mathbf{K}) = \{(x, y) \in \mathbf{K}^2 : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{\infty\},$$

onde ∞ *denota o ponto no infinito.*

Denota-se $E_{\mathbf{F}}(\mathbf{K})$ também como $E(\mathbf{K})$ definida sobre $\mathbf{F}$, isto é, com coeficientes em $\mathbf{F}$. A curvas elíptica $E_{\mathbf{F}}(\mathbf{F})$ é simplesmente denotada por $E(\mathbf{F})$.

Dada a curva elíptica $E(\mathbf{F})$ definida pela equação (4.1) os termos de Tate são definidos como:

$$\begin{array}{lll} b_3 = a_1^2 + 4a_2, & b_4 = 2a_4 + a_1a_3, & b_6 = a_1^3 + 4a_6, \\ b_8 = a_1a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2, & 4b_8 = b_2b_6 - b_4^2, & \\ c_4 = b_2^2 - 24b_4, & c_6 = -b_2 + 36b_2b_4 - 216b_6. & \end{array}$$

Definição 4.2. *Seja* E *uma curva elíptica sobre um corpo* $\mathbf{F}$ *dada pela equação (4.1). O discriminante* $\Delta(E)$ *de* E *é*

$$\Delta(E) = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6 = 12^{-3}(c_4^3 - c_6^2).$$

Se $\Delta(E) \neq 0$ *defini-se também o* j*-invariante* $j(E)$ *de* E *por*

$$j(E) = \frac{(b_2^2 - 24b_4)^3}{\Delta(E)} = \frac{12^3c_4^3}{c_4^3 - c_6^2}.$$

Dado um corpo $\mathbf{F}$ e um ponto $P = (x_0, y_0) \in \mathbf{F}^2$ uma reta de $\mathbf{F}^2$ passando por P é um conjunto da forma $r = \{(x, y) = (at + x_0, bt + y_0), \in \mathbf{F}^2 : t \in \mathbf{F}, a, b \in \mathbf{F} \text{ Fixos}\}$

Definição 4.3. *Seja* $E = E(\mathbf{F})$ *uma curva elíptica dada por (4.1). Dize-se que um ponto* $P = (x_0, y_0) \in E$ *é um ponto de singularidade se a equação*

$$(bt+y_0)^2+a_1(at+x_0)(bt+y_0)+a_3(bt+y_0) = (at+x_0)^3+a_2(at+x_0)^2+a_4(at+x_0)+a_6 \tag{4.2}$$

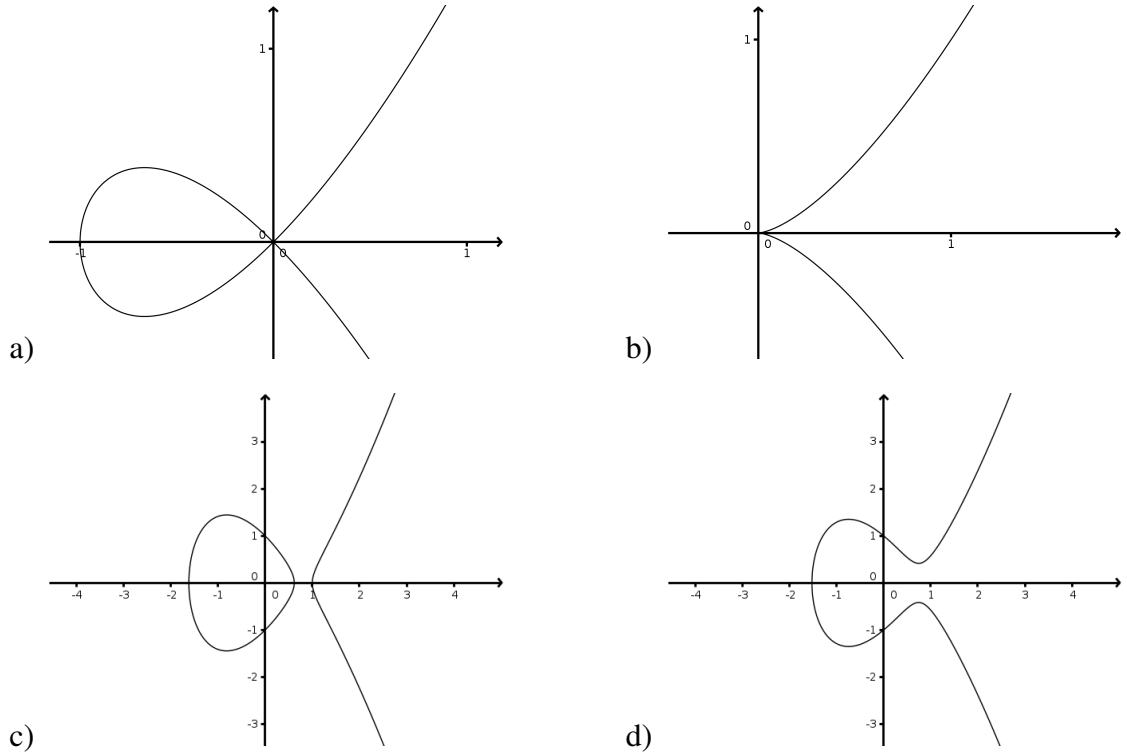
possui uma raiz dupla em t *para quaisquer* $a, b \in \mathbf{F}$*. Uma curva elíptica* $E(\mathbf{F})$ *que não possui pontos de singularidade em* $E(\mathbf{K})$ *para qualquer extensão* $\mathbf{K}|\mathbf{F}$ *é dita não-singular,*

caso contrário é dita singular.

Note que as raízes da equação (4.2) determinam os pontos da interseção entre a reta $r = \{(x, y) = (at + x_0, bt + y_0), \in \mathbf{F}^2 : t \in \mathbf{F}, a, b \in \mathbf{F} \text{ fixos}\}$ com a curva elíptica $E(\mathbf{F})$, assim, uma outra forma de dizer que P é um ponto de singularidade de E é dizer que P é uma interseção de multiplicidade maior que 1 de E com qualquer reta passando por P.

Exemplo 4. Seja $E(\mathbf{F})$ a curvas elíptica dada pela equação $y^2 = x^3 + x^2$. Qualquer reta r passando pelo ponto $(0, 0)$ é da forma $r = \{(x, y) \in \mathbf{F}^2 : x = at, y = bt, t \in \mathbf{F} \text{ e } a, b \in \mathbf{F} \text{ fixos}\}$, assim a interseção de r com $E(\mathbf{F})$ é dada pela solução da equação $(bt)^2 = (at)^3 + (at)^2$, isto é, $t^2(b^2 - a^3t - a^2) = 0$, ou seja $t = 0$ é uma raiz dupla desta equação, assim $(a0, b0) = (0, 0)$ é um ponto singular de $E(\mathbf{F})$.

Figura 4.1: Curvas elípticas sobre $\mathbb{R}$



(a) $E : y^2 = x^3 + x^2$, $\Delta(E) = 0$, singular; (b) $E : y^2 = x^3$, $\Delta(E) = 0$, singular; (c) $E : y^2 = x^3 + 2x + 1$, $\Delta(E) = -944$, não-singular; (d) $E : y^2 = x^3 + 5x/3 + 1$, $\Delta(E) = -\dfrac{19664}{27}$, não-singular.

Teorema 4.4. *Uma curva elíptica $E = E(\mathbf{F})$ sobre um corpo $\mathbf{F}$ é singular se, e somente se $\Delta(E) = 0$.*

Demonstração: Ver [18], proposição 1.4. □

A curvas elípticas trabalhadas a partir de então serão consideradas todas não-singulares.

Definição 4.5. *Duas curvas elípticas E_1 e E_2 definidas sobre um corpo* $\mathbf{F}$ *são isomorfas se existem* $u, r, s, t \in \mathbf{F}$ *com* $u \neq 0$ *tais que, para qualquer ponto* $(x, y) \in E_1$*,*

$$(x', y') = (u^2x + r, u^3y + u^2sx + t) \in E_2,$$

e para qualquer ponto $(x', y') \in E_2$*,*

$$(x, y) = (u^{-2}(x' - r), u^{-3}(y' - sx' + rs - t)) \in E_1$$

Teorema 4.6. *Sejam E_1 e E_2 curvas elípticas sobre um mesmo corpo* $\mathbf{F}$*. $j(E_1) = j(E_2)$ se, e somente se, $E_1(\mathbf{L})$ e $E_2(\mathbf{L})$ são isomorfas em alguma extensão* $\mathbf{L}|\mathbf{F}$*.*

Demonstração: Ver [21], teorema 2.19. □

4.1 ADIÇÃO DE PONTOS

Seja E uma curva elíptica sobre um corpo $\mathbf{F}$ definida pela equação (4.1). Se $P = (x_0, y_0)$ é um ponto de E defini-se seu simétrico por $-P = (x_0, -y_0 - a_1x_0 - a_3)$ e $-\infty = \infty$. Se P_1 e P_2 são dois pontos de E a adição $P_3 = P_1 + P_2$ é definida pelo algoritmo 4.7 abaixo.

Algoritmo 4.7. *Adição de pontos em uma curva elíptica*

Entrada*: Dois pontos $P_1, P_2 \in E(F)$*
Saída*: Um ponto $P_3 = P_1 + P_2 \in E(F)$*

1. *Se $P_1 = \infty$, $P_3 \leftarrow P_2$;*
2. *Se $P_2 = \infty$, $P_3 \leftarrow P_1$;*
3. *Se $P_2 = -P_1$, $P_3 \leftarrow \infty$;*
4. *Seja $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$;*
5. *Se $P_1 \neq P_2$, $\lambda \leftarrow \dfrac{y_2 - y_1}{x_2 - x_1}$, se não $\lambda \leftarrow \dfrac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}$*
6. $x_3 \leftarrow \lambda^2 + a_1\lambda - a_2 - x_1 - x_2$;
7. $y_3 \leftarrow a_1x_2 - a_3 - \lambda(x_3 - x_1) - y_1$;
8. $P_3 \leftarrow (x_3, y_3)$.

Em termos computacionais, pode-se estimar o tempo gasto para executar o algoritmo 4.7. Em um corpo cuja a ordem seja suficientemente grande, o custo de uma multiplicação ou divisão é muito superior ao custo da adição e da subtração. Isto é verdade para o corpo dos números reais em sua representação decimal e em corpos finitos $\mathbb{F}_q$, desta forma

pode-se considerar o custo do algoritmo acima apenas referente as multiplicações e divisões existentes.

Para fazer uma divisão da forma x/y em um corpo $\mathbf{F}$, é necessário encontrar o inverso do elemento y e multiplicá-lo por x, neste caso a divisão tem o custo de uma inversão, aqui representado por $\mathcal{I}$, mais uma multiplicação, representado por $\mathcal{M}$. Supondo que os passo 1, 2 e 3 não são verificados então o algoritmo irá executar os passo 4, 5, 6 e 7. O passo 4 possui um custo desprezível para se levar em consideração. O passo 5 será necessário uma multiplicação (desconsiderando multiplicações por 2 e 3) e uma inversão se $P_1 \neq P_2$ ou de 5 multiplicações e uma inversão se $P_1 = P_2$. Nos passos 6 e 7 o custo é de 2 multiplicações em cada um. Em resumo o custo do algoritmo 4.7 é de

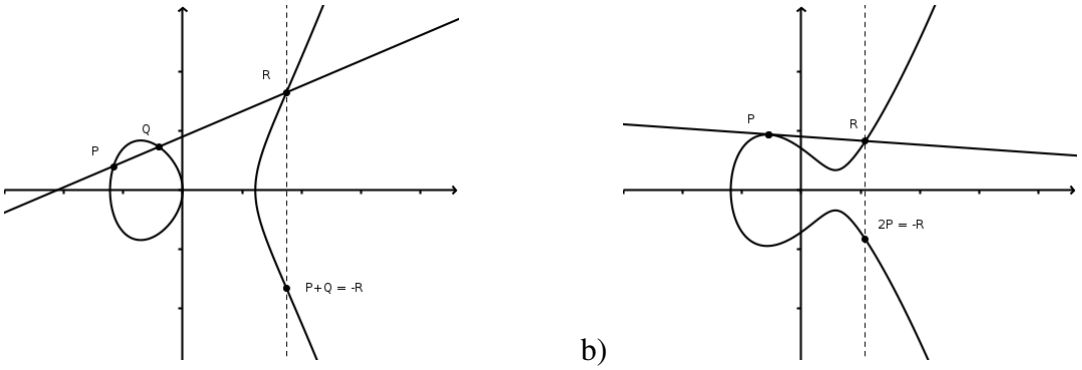
$$9\mathcal{M} + \mathcal{I} \tag{4.3}$$

para a duplicação de pontos e

$$5\mathcal{M} + \mathcal{I} \tag{4.4}$$

para a soma de pontos distintos.

Figura 4.2: Adição de pontos em curvas elípticas não-singulares sobre $\mathbb{R}$



a) b)

A figura 4.2 mostra a importância de a curva elíptica ser não-singular. Observando uma das curvas das figuras 4.1(a) ou 4.1(b) vê-se que não é possível determinar a duplicação do ponto singular da curva.

Teorema 4.8. *O conjunto dos pontos de uma curva elíptica E não-singular sobre um corpo* $\mathbf{F}$ *com a adição definida no algoritmo 4.7 é um grupo abeliano.*

Demonstração: Ver [21], teorema 2.1. □

O grupo de pontos de curvas elípticas sobre corpos finitos possui as seguintes propriedades.

Teorema 4.9 (Hasse)**.** *Se E é uma curva elíptica sobre um corpo finito $\mathbb{F}_q$, então*

$$\#(E) = q + 1 - t$$

onde $|t| \leq 2\sqrt{q}$.

Demonstração: Ver [18], teorema 1.1 ou [21], teorema 4.2. □

Teorema 4.10. *Se E é uma curva elíptica sobre um corpo finito $\mathbb{F}_q$, então*

$$E(\mathbb{F}_q) \cong \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$$

onde $n_2 \mid n_1$ e $n_1 \mid q$

Demonstração: Ver [21]. □

4.2 ENDOMORFISMO DE FROBENIUS

Definição 4.11 (Endomorfismo)**.** *Seja E uma curva elíptica sobre um corpo* **F** *definida pela equação (4.1). Dize-se que a aplicação $\alpha : E \to E$ é um endomorfismo se*

$$\alpha(P_1 + P_2) = \alpha(P_1) + \alpha(P_2)$$

para quaisquer pontos $P_1, P_2 \in E$.

Exemplos de endomorfismos são as multiplicações por escalar $n : P \mapsto nP$ onde $n \in \mathbb{Z}$. Em particular $0 : P \mapsto \infty$ e $1 : P \mapsto P$ são ditos o endomorfismo identicamente nulo e identidade respectivamente.

Se α_1 e α_2 são endomorfismos sobre uma curva elíptica E, então $\alpha_1 + \alpha_2$ e $\alpha_1 \circ \alpha_2$ são endomorfismos onde:

$$(\alpha_1 + \alpha_2)(P) = \alpha_1(P) + \alpha_2(P)$$

e

$$(\alpha_1 \circ \alpha_2)(P) = \alpha_1(\alpha_2(P))$$

para todo $P \in E$. Com estas operações de adição $\alpha_1 + \alpha_2$ e da composição $\alpha_1 \circ \alpha_2$ o conjunto dos endomorfismos sobre $E = E(\mathbf{F})$, denotado por $End(E)$, é um anel com identidade.

Se E é uma curva elíptica e $\alpha \neq n \in \mathbb{Z}$ é um endomorfismo , então dize-se que E possui multiplicação complexa. Mais que simplesmente um anel, a estrutura dos endomorfismos de uma curva elíptica poder ser uma ordem sobre o corpo $Q(\sqrt{d})$ ou em $Q(i, j, k)$, para isto segue a definição:

Definição 4.12. *Uma ordem sobre o corpo quadrático $\mathbb{Q}(\sqrt{d})$ (ou sobre $\mathbb{Q}(i,j,k)$), é um subconjunto $\mathcal{O}$ de $\mathbb{Q}(\sqrt{d})$ (respectivamente de $Q(i,j,k)$) tal que:*

i) $1 \in \mathcal{O}$;

ii) $a \in \mathcal{O} \Rightarrow ma \in \mathcal{O}$ *para todo* $m \in \mathbb{Z}$;

iii) $a, b \in \mathcal{O} \Rightarrow a + b, ab \in \mathcal{O}$;

iv) $\mathcal{O}$ *possui 2 (respectivamente 4) elementos linearmente independentes sobre* $\mathbb{Q}$.

Teorema 4.13. *Seja $E = E(\mathbf{F})$ uma curva elíptica definida sobre um corpo* **F**. *O anel de endomorfismos $End(E)$ pode satisfazer um dos seguintes casos*

i) $End(E) \cong \mathbb{Z}$;

ii) $End(E) \cong$ *Ordem no corpo quadrático* $\mathbb{Q}(\sqrt{d})$;

iii) $End(E) \cong$ *Ordem em um anel de quatérnios* $\mathbb{Q}(i,j,k)$;

Demonstração: Ver [2], página 11. □

O teorema acima mostra quais estruturas podem possuir um anel de endomorfismos de uma curva elíptica. É inportante ressaltar que o item (iii) do teorema só acontece em corpos $\mathbb{F}_p$ onde p é primo.

Seja $E(\mathbb{F}_q)$ uma curva elíptica sobre o corpo finito de ordem q e $\mathbb{F}_{q^k}$ a extensão de $\mathbb{F}_q$ de grau k. Se $\alpha : \mathbb{F}_{q^k} \to \mathbb{F}_{q^k}$ é um endomorfismo tal que $\alpha(a) = a$ para todo $a \in \mathbb{F}_q$, então a aplicação $\alpha : E(\mathbb{F}_{q^k}) \to E(\mathbb{F}_{q^k})$ onde $\alpha(x,y) = (\alpha(x), \alpha(y))$ e $\alpha(\infty) = \infty$ é um endomorfismo de curvas elípticas.

De fato, se $(x,y) \in E$ então

$$\begin{aligned} \alpha(0) &= \alpha(y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6) \\ &= (\alpha(y))^2 + a_1\alpha(x)\alpha(y) + a_3\alpha(y) - (\alpha(x))^3a_2(\alpha(x))^2 - a_4\alpha(x) - a_6 \end{aligned}$$

pois $\alpha(a_i) = a_i$, Isto significa que $(\alpha(x), \alpha(y)) \in E(\mathbb{F}_{q^k})$. A prova de que α preserva a adição de pontos pode ser feita da mesma forma substituindo P_1 e P_2 por $\alpha(P_1)$ e $\alpha(P_2)$ em cada caso do algoritmo 4.7.

Dado um corpo finito $\mathbb{F}_q$ e sua extensão $\mathbb{F}_{q^k}$ de grau k, a aplicação $\phi_q : \mathbb{F}_{q^k} \to \mathbb{F}_{q^k}$ tal que $\phi_q(x) = x^q$ é um isomorfismo de corpos que satisfaz as condições dos parágrafos anteriores, isto é, $\phi_q(a) = a$ para todo $a \in \mathbb{F}_q$. É verdade ainda que $\phi_q(a) = a$ se, e somente se, $a \in \mathbb{F}_{q^k}$.

Se $E(\mathbb{F}_{q^k})$ é uma curva elíptica com coeficientes em $\mathbb{F}_q$ o endomorfismo

$$\begin{array}{rccc} \phi_q : & \mathbb{F}_{q^k} & \to & \mathbb{F}_{q^k} \\ & (x,y) & \mapsto & (x^q, y^q) \end{array}$$

é denominado endomorfismo de Frobenius.

Como $a^{q^k} = a$ para todo $a \in \mathbb{F}_{q^k}$ a composição $\phi_q^k(x) = (\cdots((x^q)^q \cdots)^q = x^{q^k} = x$ mostra que ϕ_q é invertível cujo inverso é ϕ_q^{k-1}. Além disso o endomorfismo de Frobenius tem a seguinte propriedade:

Teorema 4.14. *Seja $E(\mathbb{F}_{q^k})$ uma curva elíptica com coeficientes em $\mathbb{F}_q$. Se $\#(E(\mathbb{F}_q)) = q + 1 - t$, então o endomorfismo de Frobenius satisfaz:*

$$\phi_q^2(P) - t\phi_q(P) + qP = \infty \tag{4.5}$$

para todo $P \in E(\mathbb{F}_{q^k})$.

Demonstração: Ver [19], teorema 4.10. □

Em resumo, o teorema acima diz que o endomorfismo obtido pela expressão $\phi_q^2 - t\phi_q + q$ é identicamente nulo. Se $\tau \in \mathbb{C}$ é uma raiz da equação $X^2 - tX + q = 0$, então pode-se identificar o endomorfismo ϕ_q com τ. Se ainda não existe $n \in \mathbb{Z}$ tal que $\phi_q(P) = nP$ para todo $P \in E(\mathbb{F}_{q^k})$, então $E(\mathbb{F}_{q^k})$ possui multiplicação complexa por ϕ_q (ou por τ). Nesta situação, a menos de isomorfismo de anéis, pode-se dizer que $\mathbb{Z}[\tau] = \{n_0 + n_1\tau : n_0, n_1 \in \mathbb{Z}\} \subset End(E)$. $\mathbb{Z}[\tau]$ é um exemplo de ordem em $\mathbb{Q}(\tau)$ como dito no teorema 4.13.

4.3 Curvas Elípticas Sobre Corpos de Característica 2, 3 e diferente de 2 e 3

Considerando a característica do corpo de definição da curvas elíptica E, há mudanças de variáveis que simplificam a equação de Weierstrass. Para uma curva elíptica $E(\mathbf{F})$ dada pela equação (4.1) sobre um corpo $\mathbf{F}$ com característica 2, pode-se considerar dois casos. Primeiro, se $a_1 \neq 0$ então a mudança de variáveis

$$(x, y) \mapsto (a_1^2 x + (a_3/a_1), a_1^3 y + a_1^{-3}(a_1 a_4 + a_3^2)) \tag{4.6}$$

transforma a equação (4.1) na equação da forma:

$$y^2 + xy = x^3 + a_2' x^2 + a_6'. \tag{4.7}$$

O discriminante de uma curva elíptica para esta equação é $\Delta(E) = a_6'$ e o j-invariante $j(E) = 1/a_6'$. Além disto, diminui-se também o custo da adição de pontos do algoritmo 4.7, nele $\lambda = (x_1^2 - y_1)/x_1$ na duplicação de pontos e $-P = -(x, y) = (x, -y - x)$.

No segundo caso, quando $a_1 = 0$, uma outra mudança de variáveis dada por

$$(x, y) \mapsto (x + a_2, y) \tag{4.8}$$

transforma (4.1) em:

$$y^2 + a'_3 y = x^3 + a'_4 x + a'_6. \tag{4.9}$$

Neste caso, tem-se $\Delta(E) = a'^4_3$ e $j(E) = 0$. O inverso de um ponto $P = (x, y) \in E$ é dado por $-P = (x, -y - a'_3)$ e $\lambda = (x_1^2 + a_4)/a_3$ na duplicação de pontos.

Em um corpo de característica 3, onde a divisão por 2 é possível, a mudança de variáveis

$$(x, y) \mapsto (x, y - (a_1 x + a_3)/2) \tag{4.10}$$

transforma a equação (4.1) na equação da forma.

$$y^2 = x^3 + a'_2 x^2 + a'_4 x + a'_6. \tag{4.11}$$

Neste caso se $E(\mathbf{F})$ é uma curva dada pela equação (4.11) e $\mathbf{F}$ um corpo de característica 3, temos $\Delta(E) = a_2^2 a_4^2 - a_2^3 a_6 - a_4^3$ e $j(E) = a_2^6/\Delta(E)$.

Nota-se que $(a_4, a_6) \neq (0, 0)$ para que $E(\mathbf{F})$ seja não-singular. A inversão de um ponto $P = (x, y) \in E(\mathbf{F})$ é $-P = (x, -y)$ e $\lambda = (2a_2 x_1 + a_4)/2y_1$ para a duplicação de pontos.

Se o corpo $\mathbf{car}(\mathbf{F}) \neq 2, 3$, a mudança de variáveis em (4.10) obtêm de (4.1) a equação (4.11), além disso, como a divisão por 3 neste corpo é possível, ainda pode-se aplicar a mudança de variáveis

$$(x, y) \mapsto (x - a'_2/3, y) \tag{4.12}$$

e transformar a equação (4.11) em:

$$y^2 = x^3 + a''_4 x + a''_6. \tag{4.13}$$

Esta última equação é chamada equação de Weierstrass na forma curta. Para uma curva elíptica definida por (4.13) tem-se $\Delta(E) = -16(4a''^3_4 + 27a''^2_6)$ e $j(E) = 12^3 \cdot 4a''^3_4/(4a''^3_4 + 27a''^2_6)$. Como em característica 3, o inverso de um ponto P é $-P = -(x, y) = (x, -y)$. No algoritmo 4.6 $\lambda = (3x_1^2 - a''_4)/2y$ na duplicação de pontos.

Em todas estas simplificações, a adição de pontos pelo algoritmo 4.7 terá um custo de $\mathcal{I} + 3\mathcal{M}$ para pontos distintos e $\mathcal{I} + 4\mathcal{M}$ para a duplicação de pontos.

Outro resultado que pode ser associado a característica do corpo de definição de curvas elípticas são as famílias de curvas a menos de isomorfismos. Em [2] é apresentado o quadro abaixo estas famílias de curvas elípticas com relação ao j-invariante sobre alguma extensão do corpo de definição.

car$p$	Discriminante	j-invariante	Curva
$p = 2$	$\Delta = a_3^4$	$j = 0$	$y^2 + a_3y = x^3 + a_4 + a_6$
$p = 2$	$\Delta = a_6$	$j = a_6^{-1}$	$y^2 + xy = x^3 + a_2x^2 + a_6$
$p = 3$	$\Delta = -a_4^3$	$j = 0$	$y^2 = x^3 + a_4x + a_6$
$p = 3$	$\Delta = -a_2^3a_6$	$j = -a_2^3a_6^{-1}$	$y^2 = x^3 + a_2x^2 + a_6$
$p \neq 2, 3$	$\Delta = -432a_6^2$	$j = 0$	$y^2 = x^3 + a_6$
$p \neq 2, 3$	$\Delta = -64a_4^3$	$j = 1728$	$y^2 = x^3 + a_4x$
$p \neq 2, 3$	$\Delta = -16(4(-a)^3 + 27(2a)^2)$	$j \neq 0, 1728$	$y^2 = x^3 - ax \pm 2a$ $a = 27j/(j - 12^3)$

4.4 SISTEMAS DE COORDENADAS

4.4.1 Plano projetivo

Seja $\mathbf{F}$ um corpo, dois pontos $(x_1, y_1, z_1), (x_2, y_2, z_2) \in \mathbf{F}^3 - \{(0, 0, 0)\}$ estão relacionados por $\sim$ se existe uma constante $k \in \mathbf{F}$ tal que $x_1 = kx_2$, $y_1 = ky_2$ e $z_1 = kz_2$. A relação $\sim$ é uma relação de equivalência em $\mathbf{F}^3 - \{(0, 0, 0)\}$, isto é

- $(x, y, z) \sim (x, y, z)$ para todo $(x, y, z) \in \mathbf{F}^3 - \{(0, 0, 0)\}$;
- Se $(x_1, y_1, z_1) \sim (x_2, y_2, z_2)$, então $(x_2, y_2, z_2) \sim (x_1, y_1, z_1)$;
- Se $(x_1, y_1, z_1) \sim (x_2, y_2, z_2)$ e $(x_2, y_2, z_2) \sim (x_3, y_3, z_3)$, então $(x_1, y_1, z_1) \sim (x_3, y_3, z_3)$.

A classe de equivalência de um ponto $(x_0, y_0, z_0) \in \mathbf{F}^3 - \{(0, 0, 0)\}$ dada pela relação $\sim$ é o conjunto:

$$(x_0 : y_0 : z_0) = \{(x, y, z) \in \mathbf{F}^3 - \{(0, 0, 0)\} : (x, y, z) \sim (x_0, y_0, z_0)\}.$$

Definição 4.15. *Dado um corpo* $\mathbf{F}$ *e a relação de equivalência* $\sim$*, o plano projetivo de* $\mathbf{F}$ *como o conjunto*

$$\mathbb{P}_{\mathbf{F}} = \{(x : y : z) : (x, y, z) \in \mathbf{F}^3 - \{(0, 0, 0)\}\}.$$

Um elemento de $\mathbb{P}_{\mathbf{F}}$ *é chamado simplesmente de ponto.*

Cada ponto (x, y) do plano cartesiano $\mathbf{F}^2$ é relacionado com o ponto $(x : y : 1)$ do plano projetivo $\mathbb{P}_{\mathbf{F}}$ e cada ponto $(x : y : z)$ com $z \neq 0$ é relacionado com $(x/z, y/z)$ em $\mathbf{F}^2$. Seja E uma curva elíptica sobre um corpo $\mathbf{F}$ definida pela equação (4.1), então para cada ponto $P = (x, y) \in E$ suas coordenadas projetivas são $(x : y : 1)$. A equação de E em coordenadas projetivas é:

$$y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz^2 + a_6z^3. \tag{4.14}$$

É fácil de verificar.

Proposição 4.16. *Se $P = (x, y)$ é uma solução de (4.1), então qualquer representante de $(x : y : 1)$ é solução de (4.14)*

Em outras palavras, a proposição 4.16 diz que as coordenadas projetivas de um ponto sobre um curva elíptica E é bem definido pela equação (4.14). Note que a equação (4.14) possui o ponto $(0 : 1 : 0)$ como solução independente de seus coeficientes, este ponto em coordenadas projetivas faz o papel do ponto no infinito em coordenadas cartesianas. O teorema abaixo relaciona pontos de coordenadas cartesianas com projetivas.

Teorema 4.17. *Seja E uma curva elíptica sobre um corpo $\mathbf{F}$ dada pela equação (4.1). Se E_1 é o conjunto dos pontos de E em coordenadas cartesianas e E_2 o conjunto dos pontos de E em coordenadas projetivas, então $\alpha : E_1 \to E_2$ tal que $\alpha(x, y) = (x : y : 1)$ e $\alpha(\infty) = (0 : 1 : 0)$ é uma função bijetora.*

Demonstração: Ver [21], página 42.

□

Como no plano cartesiano, há no plano projetivo transformações de variáveis que tornam a equação geral da curva elíptica mais simplificadas, anulando os coeficientes a_1 e a_3 para corpos com característica diferente de 2 e a_1, a_3 e a_2 para corpos com característica diferente de 2 e 3, entretanto estas transformações não serão aqui abordadas.

Seja $\mathbf{F}$ um corpo tal que $\mathbf{car}(\mathbf{F}) \neq 2, 3$ e seja $E = E(\mathbf{F})$ uma curva elíptica sobre $\mathbf{F}$ dada pela equação $y^2z = x^3 + a_4xz^2 + a_6z^3$ em coordenadas projetivas. Se $P_1 = (x_1 : y_1 : z_1)$ e $P_2 = (x_2 : y_2 : z_2)$ são pontos de E definimos a adição de $P_1 + P_2 = P_3 = (x_3 : y_3 : z_3)$ pelo algoritmo abaixo:

Algoritmo 4.18. *Adição de pontos em uma curva elíptica no Plano Projetivo*

Entrada: *Dois pontos $P_1, P_2 \in E(\mathbf{F})$*
Saída: *Um ponto $P_3 = P_1 + P_2 \in E(\mathbf{F})$*

1. *Se $P_1 = -P_2$, coloque $P_3 \leftarrow (0 : 1 : 0)$ e pare;*
2. *Se $P_1 \neq P_2$*
    3. *$u \leftarrow y_2z_1 - y_1z_2$, $v \leftarrow x_2z_1 - x_1z_2$ e $w \leftarrow u^2z_1z_2 - v^3 - 2v^2x_1z_2$;*
    4. *$x_3 \leftarrow vw$, $y_3 \leftarrow u(v^2x_1z_2 - w) - v^3y_1z_2$ e $z_3 \leftarrow v^3z_1z_2$;*
    5. *Coloque $P_3 \leftarrow (x_3 : y_3 : z_3)$ e pare;*
6. *Se não*
    7. *$t \leftarrow a_4z_1^2 + 3x_1^2$, $u \leftarrow y_1z_1$, $v \leftarrow ux_1y_1$ e $w \leftarrow t^2 - 8v$;*
    8. *$x_3 \leftarrow 2uw$, $y_3 \leftarrow t(4v - w)$ e $z_3 \leftarrow 8u^3$;*
    9. *Coloque $P_3 \leftarrow (x_3 : y_3 : z_3)$ e pare;*

Proposição 4.19. *Seja E uma curva elíptica sobre um corpo $\mathbf{F}$ tal que $\mathbf{car}(\mathbf{F}) \neq 2, 3$ dada pela equação $y^2z = x^3 + a_4xz^2 + a_6z^3$ em coordenadas projetivas. Se E_1 é o conjunto dos pontos de E em coordenadas cartesianas e E_2 o conjunto dos pontos de E em coordenadas projetivas, então $\alpha : E_2 \to E_1$ tal que $\alpha(x : y : z) = (x/z, y/z)$ se $z \neq 0$ e $\alpha((0 : 1 : 0)) = \infty$ é uma isomorfismo de grupos.*

Demonstração: Ver [21], página 42. □

Não é surpresa o resultado afirmado pela proposição anterior quando leva-se em conta que o algoritmo 4.18 foi escrito justamente para que as coordenadas cartesianas e projetivas só se diferenciem pela representação, mas representem a mesma curvas elíptica.

No algoritmo 4.18 não é necessário fazer inversão no corpo onde que a curvas estão definidas. Em [21] é afirmado que o custo deste algoritmo é de $14\mathcal{M}$ para a adição de pontos distintos e $12\mathcal{M}$ na duplicação de pontos.

4.4.2 Coordenadas Jacobianas

Seja $\mathbf{F}$ um corpo. Dois pontos $(x_1, y_1, z_1), (x_2, y_2, z_2) \in \mathbf{F}^3 - \{(0, 0, 0)\}$ estão relacionados pela relação $\sim_j$ se existe $k \in \mathbf{F}$ tal que $x_1 = k^2x_2$, $y_1 = k^3y_2$ e $z_1 = kz_2$. É fácil provar que "$\sim_j$" é uma relação de equivalência. A classe de equivalência de um ponto (x, y, z) é denotado por $(x : y : z)_j$.

Seja $\mathbf{F}$ um corpo com $\mathbf{car}(\mathbf{F}) \neq 2, 3$. As coordenadas jacobianas de um ponto $(x, y) \in \mathbf{F}^2$ é dada por $(x : y : 1)_j$ e $(x : y : z)_j$ com $z \neq 0$ são as coordenadas jacobianas do ponto $(x/z^2, y/z^3) \in \mathbf{F}^2$. Nesta mudança de coordenadas a equação de Weierstrass (4.13) se torna

$$y^2 = x^3 + a_4xz^4 + a_6z^6. \tag{4.15}$$

Neste sistema de coordenadas o ponto do infinito é representado por $(1 : 1 : 0)_j$.

Seja E é uma curva elíptica sobre um corpo $\mathbf{F}$ de característica diferente de 2 e 3. Para dois pontos $P_1 = (x_1 : y_1 : z_1)$, $P_2 = (x_2 : y_2 : z_2)$ de E em coordenadas jacobianas definimos a adição $P_1 + P_2 = P_3$ pelo algoritmo abaixo.

Algoritmo 4.20. *Adição de pontos em uma curva elíptica em Coordenadas Jacobianas*

Entrada: *Dois pontos* $P_1, P_2 \in E(\mathbf{F})$

Saída: *Um ponto* $P_3 = P_1 + P_2 \in E(\mathbf{F})$

1. Se $P_1 = -P_2$, *coloque* $P_3 \leftarrow (1:1:0)$ *e pare;*

2. Se $P_1 \neq P_2$

3. $r \leftarrow x_1 z_2^2$, $s \leftarrow x_2 z_1^2$, $t \leftarrow y_1 z_2^3$, $u \leftarrow y_2 z_1^3$, $v \leftarrow s - r$ *e* $w \leftarrow u - t$;

4. $x_3 \leftarrow -v^3 - 2rv^2 + w^2$, $y_3 \leftarrow -tv^3 + (rv^2 - x_3)w$ *e* $z_3 \leftarrow vz_1z_2$;

5. Coloque $P_3 \leftarrow (x_3 : y_3 : z_3)$ *e pare;*

6. Se não

7. $v \leftarrow 4x_1y_1^2$ *e* $w \leftarrow 3x_1^2 + a_4''z_1^4$;

8. $x_3 \leftarrow -2v + w^2$, $y_3 \leftarrow -8y_1^4 + (v = x_3)w$ *e* $z_3 \leftarrow 2y_1z_1$;

9. coloque $P_3 \leftarrow (x_3 : y_3 : z_3)$ *e pare;*

Proposição 4.21. *Seja* E *uma curva elíptica sobre um corpo* $\mathbf{F}$ *tal que* $\mathbf{car}(\mathbf{F}) \neq 2, 3$ *dada pela equação*

$$y^2 = x^3 + a_4xz^4 + a_6z^6 \tag{4.16}$$

em coordenadas jacobianas. Se E_1 *é o conjunto dos pontos de* E *em coordenadas cartesianas e* E_2 *o conjunto dos pontos de* E *em coordenadas jacobianas, então* $\alpha : E_2 \to E_1$ *tal que* $\alpha(x:y:z) = (x/z^2, y/z^3)$ *se* $z \neq 0$ *e* $\alpha((1:1:0)) = \infty$ *é uma isomorfismo de grupos.*

Demonstração: Ver [21], página 43. □

Como em coordenadas projetivas, neste algoritmo há uma melhor eficiência na adição de pontos de uma curva elíptica. Neste sistema de coordenadas, a adição de pontos distintos possui um custo de $16\mathcal{M}$ e a duplicação em $9\mathcal{M}$.

4.4.3 Coordenadas de Edwards

Teorema 4.22. *Seja* $\mathbf{F}$ *um corpo tal que* $\mathbf{car}(\mathbf{F}) \neq 2$. *Seja* $c, d \in F$ *tais que* $c, d \neq 0$ *e* d *não é um quadrado em* $\mathbf{F}$. *A curva*

$$C : u^2 + v^2 = c^2(1 + du^2v^2) \tag{4.17}$$

é isomorfa a curva elíptica

$$E : y^2 = (x - c^4d - 1)(x^2 - 4c^4d) \tag{4.18}$$

pela mudança de variáveis

$$(x, y) \mapsto \left(\frac{-2c((c^2du^2 - 1)v - c)}{u^2}, \frac{4c^2((c^2du^2 - 1)v - c) + 2c(c^4d + 1)u^2}{u^3} \right). \tag{4.19}$$

O ponto $(0, c)$ *é o elemento neutro de* C, $-(u, v) = (-u, v)$ *e*

$$(u_1, v_1) + (u_2, v_2) = \left(\frac{u_1 v_2 + u_2 v_1}{c(1 + d u_1 u_2 v_1 v_2)}, \frac{v_1 v_2 - u_1 u_2}{c(1 + d u_1 u_2 v_1 v_2)} \right) \tag{4.20}$$

Demonstração: Ver [21], teorema 2.18. □

Nestas coordenadas tanto a adição de pontos distintos como a duplicação possuem um custo de $11\mathcal{M}$.

4.5 CRIPTOGRAFIA DE CURVA ELÍPTICAS

Dado um corpo finito $\mathbb{F}_q$ e a equação (4.1), pode-se usar o algoritmo de criptografia de ELGamal sobre o grupo de pontos da curva elíptica $E(\mathbb{F}_q)$ para desenvolver um sistema de criptografia de chave pública. Pré-codificando uma mensagem dada em um ponto $M \in E(\mathbb{F}_q)$, é possível usar a estrutura de grupo do conjuntos dos pontos de $E(\mathbb{F}_q)$ para codificar M.

Dada a curva elíptica $E = E(\mathbb{F}_q)$ são necessários para o algoritmo de criptografia de ELGamal sobre E um ponto $P \in E$ e dois inteiros a e b. Escolhido estes parâmetros pelos interlocutores e com a mensagem M em mãos, calcula-se aP e usa-se o seguinte algoritmo para codificar M:

Algoritmo 4.23. *ElGamal sobre Curvas Elípticas(codificação)*

Entrada*:* $b \in \mathbb{Z}$, $M, aP \in E(\mathbb{F}_q)$
Saída*: Dois pontos* $C_1, C_2 \in E(\mathbb{F}_q)$

1. Coloque $C_1 \leftarrow M + b(aP)$*;*
2. coloque $C_2 \leftarrow bP$ *e pare;*

Dado o par de pontos C_1 e C_2 da codificação, o processo contrário, ou seja, a decodificação, é feita usando o parâmetro a pelo seguinte algoritmo.

Algoritmo 4.24. *ElGamal sobre Curvas Elípticas(decodificação)*

Entrada*:* $a \in \mathbb{Z}$, $C_1, C_2 \in E(\mathbb{F}_q)$
Saída*:* $M \in E(\mathbb{F}_q)$

1. Coloque $M \leftarrow C_1 - aC_2$ *e pare;*

A identidade $C_1 - aC_2 = (M + b(aP)) - a(bP) = M + abP - abP = M$ mostra que os algoritmos 4.23 e 4.24 são inversos um do outro. A tabela 4.1 resume o diálogo entre um receptor e um emissor utilizando a criptografia de curvas elípticas.

Como no caso geral para o algoritmo de ElGamal, decifrar o código de criptografia de curvas elípticas implica em resolver um DHP ou o DLP $xP = Q$ onde $Q = aP$.

Tabela 4.1: Algoritmo de Criptografia de Curvas Elípticas

Parâmetros Públicos	
Uma curvas elíptica $E = E(\mathbb{F}_q)$ com ordem relativamente grande e um ponto $P \in E$ com ordem N também relativamente grande	
Receptor	**Emissor**
Criação da chave privada	
Escolhe a chave privada $1 < a < N$ Calcula $A = aP$ Torna público a chave A	
Codificação	
	Escolhe aleatoriamente $1 < b < N$ Usa a chave pública de Alice A para calcular $C_1 = bP$ e $C_2 = M + bA$ Envia (C_1, C_2) para Alice
Decodificação	
Calcula $C_2 - aC_1 = M$.	

Para curvas elípticas este problema é denominado ECDLP (Elliptic Curve Discrete Logarithm Problem). O ECDLP possui uma dificuldade adicional, pois a adição de pontos de uma curvas elíptica E possui muitas operações com relação ao corpo onde esta está definida.

O teorema 4.9 garante que o grupo de pontos da curvas elíptica terá ordem suficientemente grande conforme a escolha de $\mathbb{F}_q$. O teorema 4.10 por sua vez nos garante que haverá um ponto P também de ordem suficientemente grande.

Além disso, a ordem de P não deve possuir uma fatoração em pequenos primos, entretanto a fatoração é um problema difícil de ser resolvida. Uma forma de contornar este problema é procurar por curvas elípticas que tenham ordem prima ou cuja ordem não possua fatores primos relativamente pequenos. O algoritmo de Schoof [15] é uma alternativa para a obtenção da ordem dos pontos de curvas elípticas, entretanto, este algoritmo possui um custo de $O(ln^6 q)$, um tanto lento na prática.

Outra forma de resolver este problema é determinar a ordem da curva elíptica antes mesmo de determinar seus coeficientes, isto pode ser feito pelo método da *Multiplicação Complexa - CM* (ver [3]). Entretanto não é qualquer inteiro que possar fornecer uma ordem para uma curvas elípticas. Um trabalho utilizando o método CM pode ser encontrado em [6].

4.6 ALGORITMO MOV

Definição 4.25 (torção)**.** *Seja $E = E(\mathbf{F})$ uma curva elíptica e n um inteiro positivo. Um ponto $P \in E$ é um ponto de n-torção se $nP = \infty$. O conjunto dos pontos de n-torção de um curva elíptica E é denotado por $E[n]$.*

Teorema 4.26. *Seja* $E = E(\mathbb{F}_q)$ *uma curvas elíptica e* m *um inteiro positivo tal que* $\mathbf{mdc}(m, q) = 1$*, então existe um inteiro* $k \geq 1$ *tal que*

$$E(\mathbb{F}_{q^k})[m] \cong \mathbb{Z}_m \times \mathbb{Z}_m$$

Demonstração: Ver [11]. □

Dado um inteiro m e uma curvas elíptica $E(\mathbb{F}_q)$, o *grau de imersão* de E relativo a m é o menor inteiro k tal que $E(\mathbb{F}_{q^k})[m] \cong \mathbb{Z}_m \times \mathbb{Z}_m$.

Definição 4.27 (Emparelhamento de Weil)**.** *Sejam uma curvas elíptica* $E = E(\mathbb{F}_q)$ *e um inteiro positivo* m *tal que* $\mathbf{mdc}(m, q) = 1$*. O emparelhamento de Weil* e_m *é a função*

$$e_m : E[m] \times E[m] \to \mu_m \tag{4.21}$$

onde $\mu_m = \{u \in \overline{\mathbb{F}_q} : u^m = 1\}$ *tal que:*

1. *Para todo* $P_1, P_2, Q \in E[m]$*,* $e_m(P_1 + P_2, Q) = e_m(P_1, Q)e_m(P_2, Q)$*;*
2. *para todo* $P, Q_1, Q_2 \in e[m]$*,* $e_m(P, Q_1 + Q_2) = e_M(P, Q_1)e_m(P, Q_2)$*;*
3. *para todo* $P \in E[m]$*,* $e_m(P, P) = 1$*;*
4. *se para todo* $Q \in E[m]$*,* $e_m(P, Q) = 1$*, então* $P = \infty$*.*

O emparelhamento de Weil pode ser implementado trabalhando-se com funções racionais proposto por Miller em [12]. O custo para aplicar o emparelhamento de Weil sobre uma curva elíptica $E(\mathbb{F}_q)$ é $O(\log_2 m)$. Este emparelhamento é base para o seguinte algoritmo:

Algoritmo 4.28. *MOV*

Entrada*: Dois pontos* $P, Q \in E(\mathbb{F}_q)$
Saída*: Um inteiro* n *tal que* $Q = nP$

1. *Se* l *é a ordem de* P*, encontre o grau de imersão* k *de* E *relativo a* l*;*
2. *calcule* $N = \#E(\mathbb{F}_{q^k})$*;*
3. *encontre um ponto* $T \in E(\mathbb{F}_{q^k})$ *de modo aleatório;*
4. *calcule* $T' = (N/l)T$*. Se* $T' = \infty$ *volte ao passo 3;*
5. *calcule* $\alpha = e_m(P, T')$ *e* $\beta = e_m(Q, T')$*;*
6. *encontre* n *tal que* $\alpha^n = \beta$ *e pare.*

Neste algoritmo, apesar dos passos de 1 a 5 serem lentos, estes possui um custo polinomial. O passo 6 é o único que possui um custo maior. Como visto, o custo do DLP sobre corpos finitos é subexponencial, entretanto a entrada deste algoritmo é uma curva elíptica

sobre $\mathbb{F}_q$, e não o corpo $\mathbb{F}_{q^k}$ onde o DLP será resolvido. Se $k > \ln^2 q$, o custo deste algoritmo será exponencial. Alguns casos onde isso não ocorre são para uma família de curva elípticas denominadas super-singulares. A definição de tais curvas é dada abaixo.

Definição 4.29. *Seja $E(\mathbb{F}_q)$ uma curva elíptica e $\#E(\mathbb{F}_q) = q + 1 - t$. Dizemos que E é super-singular se $\mathbf{car}(\mathbb{F}_q) \mid t$*

Teorema 4.30. *Seja $E = E(\mathbb{F}_q)$ uma curva elíptica e $P \in E$ um ponto de ordem m. Se E é super-singular, então o grau de imersão de E relativo a m é menor ou igual a 6*

Demonstração: Ver [11], seção IV.

□

5 CURVAS ELÍPTICAS SOBRE $\mathbb{F}_{q^k}$ COM COEFICIENTES EM $\mathbb{F}_q$

5.1 ORDEM DOS PONTOS DE $E(\mathbb{F}_{q^k})$

Teorema 5.1. *Seja $E(\mathbb{F}_q)$ uma curva elíptica tal que $\#E(\mathbb{F}_q) = q + 1 - t$. Se α e β são as raízes da equação $X^2 - tX + q = 0$ então*

$$\#E(\mathbb{F}_{q^k}) = q^k + 1 - (\alpha^k + \beta^k)$$

qualquer que seja $k \geq 1$.

Demonstração: Ver [21], teorema 4.12. □

Seja $E(\mathbb{F}_q)$ um curvas elíptica e t_k o inteiro tal que $\#E(\mathbb{F}_{q^k})) = q^k + 1 - t_k$. Se t_1 é conhecido, então pode-se encontrar as raízes α e β da equação $X^2 - tX + q = 0$ e obter $t_k = \alpha^k + \beta^k$ para qualquer inteiro positivo k. Como

$$\alpha = \frac{t}{2} + \frac{\sqrt{t^2 - 4q}}{2} \text{ e } \beta = \frac{t}{2} - \frac{\sqrt{t^2 - 4q}}{2},$$

tem-se

$$t_k = \left(\frac{t}{2} + \frac{\sqrt{t^2 - 4q}}{2}\right)^k + \left(\frac{t}{2} - \frac{\sqrt{t^2 - 4q}}{2}\right)^k.$$

Por exemplo

$$\begin{aligned} t_2 &= \alpha^2 + \beta^2 \\ &= \left(\frac{t}{2} + \frac{\sqrt{t^2 - 4q}}{2}\right)^2 + \left(\frac{t}{2} - \frac{\sqrt{t^2 - 4q}}{2}\right)^2 \\ &= \frac{t^2}{4} + \frac{t^2 - 4q}{4} + 2\frac{t}{2}\frac{\sqrt{t^2 - 4q}}{2} + \frac{t^2}{4} + \frac{t^2 - 4q}{4} - 2\frac{t}{2}\frac{\sqrt{t^2 - 4q}}{2} \\ &= \frac{t^2}{2} + \frac{t^2 - 4q}{2} \\ &= t^2 - 2q \end{aligned}$$

Ao invés de calcular as potências de α e β pode-se determinar a sequência $t_1, t_2, \cdots, t_k$ pela recorrência do seguinte teorema.

Teorema 5.2. *Seja $E(\mathbb{F}_q)$ uma curvas elíptica tal que $\#E(\mathbb{F}_q) = q+1-t_1$. Se $t_1, t_2, \cdots, t_k, \cdots$ é a sequência de inteiros tal que para cada inteiro positivo k $\#E(\mathbb{F}_{q^k}) = q^k + 1 - t_k$, então*

$$t_{k+2} = t_1 t_{k+1} - q t_k \tag{5.1}$$

Demonstração: Sejam α e β as raízes de $X^2 - t_1 X + q = 0$, então $\alpha + \beta = t_1$ e $\alpha\beta = q$. Além disso pelo teorema anterior $t_k = \alpha^k + \beta^k$, segue daí que

$$\begin{aligned} t_1 t_{k+1} - q t_k &= (\alpha+\beta)\left(\alpha^{k+1} + \beta^{k+1}\right) - q\left(\alpha^k + \beta^k\right) \\ &= \alpha^{k+2} + \alpha\beta^{k+1} + \beta\alpha^{k+1} + \beta^{k+2} - q\left(\alpha^k + \beta^k\right) \\ &= \alpha^{k+2} + \beta^{k+2} + \alpha\beta\left(\alpha^k + \beta^k\right) - q\left(\alpha^k + \beta^k\right) \\ &= \alpha^{k+2} + \beta^{k+2} + q\left(\alpha^k + \beta^k\right) - q\left(\alpha^k + \beta^k\right) \\ &= \alpha^{k+2} + \beta^{k+2} \\ &= t_{k+2}. \end{aligned}$$

□

Como $t_2 = t_1^2 - 2q$, os valores de t_k com $k \geq 2$ ficam bem determinados conhecendo t_1. O seguinte algoritmo utiliza a recorrência do teorema 5.2 para calcular $\#(E(\mathbb{F}_{q^k}))$ a partir de $\#(E(\mathbb{F}_q))$.

Algoritmo 5.3. *Ordem de $E(\mathbb{F}_{q^k})$ com coeficientes em $\mathbb{F}_q$*

Entrada*: A curva elíptica $E(\mathbb{F}_q)$;*
Saída*: O inteiro $N = \#E(\mathbb{F}_{q^k})$;*

1. *Calcule $\#E(\mathbb{F}_q)$;*
2. *$t \leftarrow \#E(\mathbb{F}_q) - q - 1$;*
3. *$t_1 \leftarrow t$, $t_2 \leftarrow t_1^2 - 2q$ e $i \leftarrow 1$;*
4. *se $k \leq 2$ coloque $N \leftarrow t_k$ e pare;*
5. *$N \leftarrow t t_2 - q t_1$;*
6. *se $i < k$, então $i \leftarrow i + 1$ e volte ao passo 4;*
7. *se $i = k$ pare.*

Observa-se que o passo 1 é o mais demorado para ser feito neste algoritmo, no mais só há duas multiplicação de inteiros no passo 5. Este algoritmo é no entanto eficiente quando q é pequeno, neste caso o passo 1 não faz muita diferença no tempo execução e o algoritmo se torna polinomial.

5.2 MULTIPLICAÇÃO POR ESCALAR INTEIRO

O algoritmo anterior nos mostra que podemos contornar o problema do cálculo da ordem de uma curvas elíptica se utilizarmos curvas sobre corpos $\mathbb{F}_{q^k}$ com coeficientes em $\mathbb{F}_q$. Outra vantagem obtida com esta escolha é considerar a equação $\phi_q^2(P) - t\phi_q(P) + qP = \infty$ para o endomorfismo de Frobenius do teorema 4.14. Isolando qP nesta equação tem-se $qP = t\phi_q(P) - \phi_p^2(P)$ para todo $P \in E(\mathbb{F}_{q^k})$, ou ainda escrita de outra maneira

$$q = t\phi_q - \phi_p^2. \tag{5.2}$$

Seja n um inteiro e $P \in E(\mathbb{F}_{q^k})$, se $n = n_0 + n_1 q + \cdots + n_l q^l$ é a representação de n na base q, então é possível calcular nP pela igualdade

$$nP = n_0 P + n_1 qP + \cdots + n_l q^l P. \tag{5.3}$$

Substituindo cada q desta equação por (5.2) obtêm-se uma expressão em ϕ_q. Se nesta expressão alguns dos coeficientes é maior que q podemos reescreve-lo na representação por q e novamente substituir por (5.2).

Exemplo 5. Seja $q = 5$, $t = 4$ e $n = 21$, como $5 = 4\tau - \tau^2$ tem-se

$$\begin{aligned}
21 &= 1 + 4 \cdot 5 \\
&= 1 + 4(4\tau - \tau^2) \\
&= 1 + (16 - 4\tau)\tau \\
&= 1 + (1 + 3 \cdot 5 - 4\tau)\tau \\
&= 1 + \tau + (3(4\tau - \tau^2) - 4\tau)\tau \\
&= 1 + \tau + (8 - 3\tau)\tau^2 \\
&= 1 + \tau + (3 + 5 - 3\tau)\tau^2 \\
&= 1 + \tau + (3 + (4\tau - \tau^2) - 3\tau)\tau^2 \\
&= 1 + \tau + 3\tau^2 + \tau^3 - \tau^4
\end{aligned}$$

Seja τ uma raiz de $X^2 - tX + q = 0$, no exemplo acima o que se obtêm é a representação de um inteiro n por τ utilizando a identidade $p = t\tau - \tau^2$. O método sistemático de fazer isso é o algoritmo abaixo:

Algoritmo 5.4. *Representação em* $\mathbb{Z}[\tau]$

Entrada*: Um elemento* $n = a + b\tau \in \mathbb{Z}[\tau]$ *onde* $q = t\tau - \tau^2$

Saída*: A sequência* $n_0, n_1, \cdots, n_l$ *onde* $n = n_0 + n_1\tau + \cdots + n_l\tau^l$ *onde* $|n_i| < q$

1. $i \leftarrow 0$;
2. $a = qs + r$ *com* $|r| < q$;
3. $n_i \leftarrow r$;
4. $a \leftarrow st + b$ *e* $b \leftarrow -s$;
5. $i \leftarrow i + 1$;
6. *se* $|a| + |b| \neq 0$, *então volte ao passo 2, se não, pare.*

Observe que neste algoritmo não é exigido que $r \geq 0$ no passo 2, apenas que esteja no intervalo de $(-q, q)$. Pode parecer que o passo 4 faz com que a se torne tão grande que a condição de parada no passo 6 nunca será satisfeita, entretanto deve-se ter em

consideração que $|t| \leq 2\sqrt{q}$ e isso é suficiente para que o algoritmo pare em algum momento. Dados experimentais indicam que a quantidade de vezes que o loop do algoritmo será executado não ultrapassa $2\log_q n + 1$.

No algoritmo 5.4 é feita uma divisão de inteiros no passo 2 e uma multiplicação no passo 4, se o loop deste algoritmo é executado $2\log_q n$ como dito anteriormente então seu custo é de $2\log_q n(\log^2 n + \log^2 n) = O(\log^3 n)$.

Usando a equação

$$n = n_0 + n_1\tau + \cdots + n_l\tau^l \tag{5.4}$$

obtida pelo algoritmo 5.4 com $n \in \mathbb{Z}$ e $P \in E(\mathbb{F}_q)$ e considerando que a multiplicação por τ é o endomorfismo de Frobenius o calculo de nP pode ser feito da forma

$$\begin{aligned} nP &= n_0P + n_1\phi_q(P) + \cdots + n_l\phi_q^l(P) \\ &= n_0P + \phi_q(n_1P) + \cdots + \phi_q^l(n_lP) \\ &= n_0P + \phi_q(n_1P + \phi_q(n_2P + \cdots \phi_q(n_lP)\cdots)). \end{aligned} \tag{5.5}$$

Como $-q < n_i < q$ para todo $i = 1, 2\cdots, l$, a sequência $P, 2P, 3P, \cdots, (q-1)P$ pode ser calculada antecipadamente. O algoritmo para o produto nP usando a expressão 5.5 é apresentado abaixo.

Algoritmo 5.5. *Produto nP com endomorfismo de Frobenius*

Entrada*: Um ponto $P \in E(\mathbb{F}_q)$ e um inteiro $n = n_0 + n_1\tau + \cdots n_l\tau$*
Saída*: O ponto $Q = nP \in E(\mathbb{F}_q)$*

1. Coloque $P_i \leftarrow iP$ para $i = 1, 2, \cdots, (q-1)$;
2. coloque $i \leftarrow l$;
3. se $n_l \geq 0$ coloque $Q \leftarrow P_{n_l}$, se não coloque $Q \leftarrow -P_{n_l}$;
4. $i \leftarrow i - 1$;
5. se $n_i \geq 0$ coloque $Q \leftarrow P_{n_i} + \phi_q(Q)$, se não coloque $Q \leftarrow (-P_{n_i}) + \phi_q(Q)$;
6. se $i \neq 0$ volte ao passo 4, se não pare.

Neste algoritmo são necessários $(q-1)$ adição de pontos no passo 1, $2\log_q n$ aplicação do endomorfismo de Frobenius em cada coordenada do ponto Q no passo 5 e $2\log_q n$ adição de pontos no mesmo passo.

Supondo que em média metade dos $n_{i's}$ são negativos, é necessário $(q-1)/2$ inversão de pontos juntando os passos 3 e 5, entretanto estes custos serão desconsiderados, pois em corpos de característica diferente de 2 a inversão de um ponto $P = (x, y)$ é $-P = (x, -y)$, de forma que não há grandes perdas no tempo do algoritmo.

Usando o algoritmo 2.6, cada endomorfismo de Frobenius pode ser feito a

um custo de $(2\log_2 q)$ vezes uma multiplicações no corpo $\mathbb{F}_{q^k}$. Usando o custo para a adição e duplicação de pontos em 4.4 e 4.3 o custo do algoritmo 5.5 é

$$(q-1)(5\mathcal{M}+\mathcal{I}) + 2\log_q n(4\mathcal{M}\log_2 q + 5\mathcal{M} + \mathcal{I}). \tag{5.6}$$

Onde, novamente, $\mathcal{M}$ e $\mathcal{I}$ representam respectivamente o custo de uma multiplicação e uma inversão do corpo de definição da curvas elíptica. Isto levando em consideração que dificilmente será encontrado uma duplicação de pontos no algoritmo 5.5.

Usando o algoritmo 2.6, a expressão para o custo do cálculo de nP é

$$(9\mathcal{M}+\mathcal{I})\log_2 n + 0,5(5\mathcal{M}+\mathcal{I})\log_2 n = (11,5\mathcal{M}+1,5\mathcal{I})\log_2 n. \tag{5.7}$$

Em $\mathbb{F}_{q^k}$, [3] mostra que $\mathcal{M} = O(\log_2^2 q^k)$ enquanto que $\mathcal{I} = O(\log_2^3 q^k)$. Considerando o inteiro positivo N o tamanho das chaves de criptografia em bits, isto é, $q^k \approx 2^N$, então tem-se $\mathcal{M} \approx N^2$, $\mathcal{I} \approx N^3$, $k \approx N\log_q 2$ e $n \approx 2^N$. Pelas equações (5.6) e (5.7) tem-se respectivamente:

$$(2\log_q 2)N^4 + (7 + q + 10\log_q 2)N^3 + 5(q-1)N^2, \tag{5.8}$$

$$1,5N^4 + 11,5N^3. \tag{5.9}$$

Tomando q fixo, tanto (5.8) quanto (5.9) são expressões polinomiais de N com grau 4. Embora o algoritmo 2.6 e 5.5 possuam o mesmo custo assintótico (ambos iguais a $O(N^4)$), o coeficiente principal de (5.8) é cada vez menor quanto maior for a escolha de q, isto implica que para certos valores de q e N o tempo necessário para executar 5.5 é menor que o tempo para executar 2.6. O gráfico da figura 5.1 mostra isso para valores de N iguais a 64, 128 e 256 e q variando de 3 a 100.

A figura 5.1 diz que não há grandes ganhos de eficiência na utilização do algoritmo 5.5 para 64 bits, mas para 128 e 256 bits este algoritmo possui uma melhor velocidade comparada com o algoritmo 2.6 para alguns valores de q. Embora útil para para visualizar o comportamento dos algoritmo 2.6 e 5.5, este gráfico apresenta algumas diferenças com a comparação real deste algoritmos. Isto se deve ao uso de N^2 e N^3 para representar os custos de uma multiplicação e inversão no corpo finito. Este são custos assintóticos e não descrevem precisamente o custo real destas operações.

Uma outra forma de comparar as expressões 5.8 e 5.9 é fazer sua diferença (dada por 5.10) e procurar pelos intervalos para N e q tais que esta seja negativa. Pode-se ainda encontrar o mínimo desta diferença derivando a expressão obtida e encontrando seus zeros.

$$(2\log_q 2 - 1,5)N^4 + (q - 4,5 + 12\log_q 2)N^3 + 5(q-1)N^2. \tag{5.10}$$

Figura 5.1: Tempo percentual teórico gasto pelo algoritmo 5.5 com relação ao algoritmo 2.6

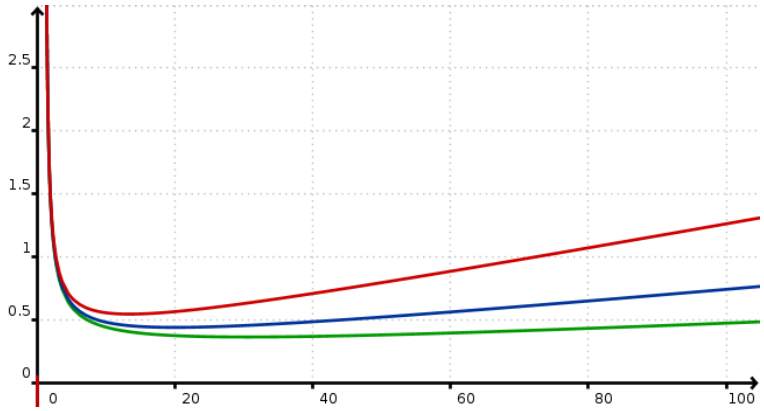


Gráfico da proporção de (5.8) para (5.9) com $N = 64$ em vermelho, $N = 128$ em azul e $N = 256$ em verde.

Derivando esta expressão com relação a q temos

$$-\frac{2\ln 2}{q\ln^2 q}N^4 + \left(1 - \frac{12\ln 2}{q\ln^2 q}\right)N^3 + 5N^2. \tag{5.11}$$

Encontrar os zeros desta última expressão de modo analítico é um tanto complicado por envolver logaritmos. Um meio para contornar esta situação é aproximar os zeros por métodos numéricos como o método de Newton [20] ou o método da bissecção. Para valores fixos de $N = 64$, 128 e 256, o método de Newton encontra os zeros de (5.11), com precisão de duas casas decimais, $q = 13{,}37$, 19,95 e 30,5 respectivamente. Como q é inteiro pode-se tomar os valores aproximados $q = 13, 20$ e 31 respectivamente.

5.3 Curvas de Koblitz

A inspiração para utilizar o endomorfismo de Frobenius na multiplicação por escalar de pontos sobre uma curva elíptica é devido as curvas de Koblitz [4] onde é usado para melhorar a eficiência da duplicação de pontos de curvas definidas sobre $\mathbb{F}_2$.

Sobre o corpo $\mathbb{F}_2$ há, a menos de isomorfismos, duas curvas não-singulares definidas pelas equações

$$E_a : y^2 + xy = x^3 + ax^2 + 1 \text{ com } a \in \{0, 1\}, \tag{5.12}$$

denominadas curvas de Koblitz. Para estas duas curvas as cardinalidades dos conjuntos de pontos são $\#(E_a) = 2 + 1 - t_a$ onde $t_0 = -1$ e $t_1 = 1$. Além disso, pelo teorema 4.14, o endomorfismo de Frobenius satisfaz $\phi_2^2(P) - t_a\phi(P) + 2P = \infty$ qualquer que seja o ponto P em $E_a(\mathbb{F}_{2^k})$.

Dado qualquer inteiro $n \in \mathbb{Z}$, o algoritmo 5.4 nos permite escrevê-lo como $n = n_0 + n_1\tau + \cdots n_l\tau^l$ onde τ é uma raiz da equação $X^2 - t_aX + 2 = 0$, $l \leq 2\log_2 n$ e $n_i \in \{-1, 0, 1\}$ para todo $i = 0, 1, \cdots, l$. Esta representação de n é semelhante a representação binária com a diferença que cada n_i pode assumir também valores negativos. Desta forma pode-se utilizar um algoritmo semelhante ao 2.6 para calcular a multiplicação nP, levando em consideração quando algum n_i é negativo e substituindo a duplicação de pontos pelo endomorfismo de Frobenius ϕ_2.

A expressão (4.3) nos dá o custo de uma duplicação de pontos por $9\mathcal{M} + \mathcal{I}$ enquanto que o custo de aplicar ϕ_2 é $2\mathcal{M}$. O algoritmo 2.6 nesta situação fará $2\log_2 n$ duplicações e $\frac{1}{2}2\log_2 n$ adições de pontos. Usando o endomorfismo de Frobenius nas curvas de Koblitz o custo que do cálculo de nP é

$$2(2\log_2 n)\mathcal{M} + \log_2 n(5\mathcal{M} + \mathcal{I}). \tag{5.13}$$

Fazendo as substituições de $\mathcal{M} = N^2$, $\mathcal{I} = N^3$ e $\log_2 n = N$ como anteriormente para o custo de 5.5 e 2.6 temos

$$N^4 + 9N^3. \tag{5.14}$$

Uma rápida comparação de (5.14) com (5.9) dá uma ideia de quão eficiente é o método de Koblitz. Observa-se ainda que o trabalho com curvas de Koblitz é um caso particular do método geral utilizando o algoritmo 5.5, embora sua eficiência com relação ao algoritmo tradicional não dependa do tamanho da chave N.

5.4 POLINÔMIOS IRREDUTÍVEIS

A questão de determinar se um número é primo ou composto sempre incentivou a comunidade matemática em direção de avanços na área da teoria dos números, mais ainda com o avanço da informática e o desenvolvimento de algoritmos de criptografia baseados nestes números. Neste trabalho a importância dos números primos não é diferente. Por exemplo, o corpo finito com p elementos, onde p é primo, pode ser representado e operado usando-se os elementos de $\mathbb{Z}_p$. Mais que isso, também existe a importância de determinar se um polinômio sobre $\mathbb{Z}_p[X]$ é irredutível, pois na prática, uma das formas de trabalhar com o corpo $\mathbb{F}_q$ onde $q = p^k$ é usar o anel quociente $\mathbb{Z}_p[X]/(f)$ onde f é um polinômio irredutível de grau k.

Se p é um número primo, então o pequeno teorema de Fermat [5] garante que para todo $a \in \mathbb{Z}_p$, $a^p = a$. Se $a \neq 0$ então é ainda verdade que $a^{p-1} = 1$ em $\mathbb{Z}_p$. Entretanto, esta propriedade não é exclusiva para números primos, por exemplo, se $a \in \mathbb{Z}_{561}$, então $a^{561} = a$, mas $561 = 3 \times 11 \times 17$. A igualdade $a^{p-1} = 1$ em $\mathbb{Z}_p$ pode ser usada da seguinte forma:

Se p um número primo e $p - 1 = 2^e r$ com r ímpar, então para todo $a \in \mathbb{Z}_p^*$,

tem-se:

$$\begin{aligned} 0 &= a^{p-1} - 1 \\ &= a^{2^e r} - 1 \\ &= (a^{2^{e-1} r} + 1)(a^{2^{e-1} r} - 1) \\ &= (a^{2^{e-1} r} + 1)(a^{2^{e-2} r} + 1)(a^{2^{e-2} r} - 1) \\ &= \cdots \\ &= (a^{2^{e-1} r} + 1)(a^{2^{e-2} r} + 1) \cdots (a^{2^e r} + 1)(a^r + 1)(a^r - 1). \end{aligned}$$

Como $\mathbb{Z}_p$ é um corpo, um dos fatores do último produto é 0, isto significa que $a^{2^i r} = -1$ para algum $0 \leq i < e$, ou $a^r = 1$. O teorema abaixo resume este fato.

Teorema 5.6. *Seja p um número primo e $p - 1 = 2^e r$ onde r é ímpar. Se $a \in \mathbb{Z}_p^*$, então uma das seguintes condições é satisfeita:*

i) $a^{2^i r} = -1$ *para algum inteiro* $0 \leq i < e$*;*

ii) $a^r = 1$.

Definição 5.7. *Seja n um inteiro composto $a \in \mathbb{Z}_n$ e $n - 1 = 2^e r$ com r ímpar. Dize-se que n é pseudo-primo para a base a se uma das seguintes condições é satisfeita:*

i) $a^{2^i r} = -1$ *para algum inteiro* $0 \leq i < e$*;*

ii) $a^r = 1$.

Teorema 5.8. *Se n um inteiro composto, então n é pseudo-primo para no máximo 25% das bases $a \in \mathbb{Z}_n$.*

Demonstração: Ver [10], proposição 2.1. □

Pode-se obter um resultado semelhante ao referente à números pseudo-primos para polinômios. Para isso seguem os resultados.

Teorema 5.9. *Seja $f \in \mathbb{F}_q[X]$ é um polinômio irredutível de grau $k \geq 1$ e $q^k - 1 = 2^e r$ onde r é ímpar. Se $a \in \left(\frac{\mathbb{F}_q[X]}{(f)}\right)^*$, então uma das seguintes condições é satisfeita:*

i) $a^{2^i r} = -1$ *para algum inteiro* $0 \leq i < e$*;*

ii) $a^r = 1$.

Demonstração: Se f é um polinômio irredutível de grau k sobre o corpo $\mathbb{F}_q$, então o corpo $\frac{\mathbb{F}_q[X]}{(f)}$ possui q^k elementos e $q^k - 1$ elementos não nulos que possui a estrutura

de grupo com a operação de multiplicação, assim para qualquer $a \in \left(\frac{\mathbb{F}_q[X]}{(f)}\right)^*$ tem-se que $a^{q^k-1} = 1$. Além disso, se $q^k - 1 = 2^e r$ com r ímpar, então

$$\begin{aligned} 0 &= a^{q^k-1} - 1 \\ &= a^{2^e r} - 1 \\ &= (a^{2^{e-1}r} + 1)(a^{2^{e-1}r} - 1) \\ &= (a^{2^{e-1}r} + 1)(a^{2^{e-2}r} + 1)(a^{2^{e-2}r} - 1) \\ &= \cdots \\ &= (a^{2^{e-1}r} + 1)(a^{2^{e-2}r} + 1)\cdots(a^{2^e r} + 1)(a^r + 1)(a^r - 1). \end{aligned}$$

E da mesma forma que $\frac{\mathbb{F}_q[X]}{(f)}$ é um corpo, um dos fatores do último produto é 0, isto significa que $a^{2^i r} = -1$ para algum $0 \leq i < e$ ou $a^r = 1$.

□

Definição 5.10. *Seja $f \in \mathbb{F}_q[X]$ um polinômio não irredutível de grau k, $q^k - 1 = 2^e r$ com r ímpar e $a \in \frac{\mathbb{F}_q[X]}{(f)}$. Dize-se que f é pseudo-primo para a base a se uma das seguintes condições é satisfeita:*

i) $a^{2^i r} = -1$ *para algum inteiro* $0 \leq i < e$*;*

ii) $a^r = 1$.

Para polinômios irredutíveis há um resultado análogo algo teorema 5.8, porem com uma diferença em relação a porcentagem.

Teorema 5.11. *Se $f \in \mathbb{F}_q[X]$ é um polinômio não-irredutível de grau $k \geq 2$, então f é pseudo-primo para no máximo 50% das bases $g \in \frac{\mathbb{F}_q[X]}{(f)}$.*

Demonstração: Seja $q^k - 1 = 2^e r$. A demonstração será dividida em dois casos.

1º caso: Suponha que exista $v \in \mathbb{F}_q[X]$ irredutível tal que $v^2 | f$.

Neste caso se $g \in \mathbb{F}_q[X]$ é tal que $g^{2^e r} \equiv 1 \mod (f)$, então é verdade também que $g^{2^e r} \equiv 1 \mod (v^2)$. Assim, basta mostrar que a porcentagem dos $g \in \mathbb{F}_q[X]/(v^2)$ tais que $g^{2^e r} = 1$ é menor ou igual a 50%.

Como $\left(\frac{\mathbb{F}_q[X]}{(v^2)}\right)^*$ possui um gerador u, tem-se que $\left(\frac{\mathbb{F}_q[X]}{(v^2)}\right)^* = \{u^i : i \in \mathbb{Z}\}$. Este conjunto possui $q^{\mathbf{gr}(v)}(q^{\mathbf{gr}(v)} - 1)$ elementos. Se $g \in \frac{\mathbb{F}_q[X]}{(v^2)}$ é tal que $g^{2^e r} = 1$, então $g \in \left(\frac{\mathbb{F}_q[X]}{(v^2)}\right)^*$, assim existe $i \in \mathbb{Z}$ tal que $g = u^i$. Observe que $(u^i)^{2^e r} \equiv u^{2^e r i} \equiv 1 \mod (v^2)$ se, e somente se, $\#\left(\left(\frac{\mathbb{F}_q[X]}{(v^2)}\right)^*\right) | 2^e r i$, como $q^{\mathbf{gr}(v)} | q^{\mathbf{gr}(f)}$ e $q^{\mathbf{gr}(f)} = 2^e r$ tem-se que $q^{\mathbf{gr}(v)} \nmid 2^e r$ de modo que $q^{\mathbf{gr}(v)} | i$. Visto que existem apenas $q^{\mathbf{gr}(v)} - 1$ i's entre 1 e

$q^{\mathbf{gr}(v)}(q^{\mathbf{gr}(v)}-1)$ divisíveis por $q^{\mathbf{gr}(v)}$, existem no máximo $q^{\mathbf{gr}(v)}-1$ elementos em $\left(\frac{\mathbb{F}_q[X]}{(v^2)}\right)^*$ (e consequentemente em $\frac{\mathbb{F}_q[X]}{(v^2)}$) satisfazendo $g^{2^{e_r}} \equiv 1 \mod (v^2)$. Consequentemente a fração dos elementos de grau menor ou igual a $\mathbf{gr}(v^2)$ satisfazendo $g^{2^{e_r}} equiv 1 \mod (v^2)$ é menor ou igual a

$$\frac{q^{\mathbf{gr}(v)}-1}{q^{\mathbf{gr}(v^2)}-1} = \frac{1}{q^{\mathbf{gr}(v)}+1} \leq \frac{1}{3} \leq \frac{1}{2}.$$

2º caso: Suponha agora que $f = v_1 v_2 \cdots v_s$ com $s \geq 2$ onde os $v_{j's}$ são polinômios irredutíveis distintos.

Para cada v_j escreva $q^{\mathbf{gr}(v_j)}-1 = 2^{e_j} r_j$ com r_j ímpar. Se f é pseudo-primo com relação à $g \in \mathbb{F}_q[X]$ onde $\mathbf{gr}(g) < \mathbf{gr}(f)$ e $q^{\mathbf{gr}(f)}-1 = 2^e r$ onde r é ímpar, então uma das seguintes condições acontece:

$$\begin{aligned} g^r &\equiv 1 \mod f \\ g^{2^i r} &\equiv -1 \mod f \text{ para algum } 0 \leq i \leq e. \end{aligned}$$

Em ambos os casos tem-se que $g \in \left(\frac{\mathbb{F}_q[X]}{(f)}\right)^*$. O teorema chinês do resto garante o isomorfismo:

$$\begin{array}{rccc} \varphi: & \left(\frac{\mathbb{F}_q[X]}{(f)}\right)^* & \to & \left(\frac{\mathbb{F}_q[X]}{(v_1)}\right)^* \times \cdots \times \left(\frac{\mathbb{F}_q[X]}{(v_s)}\right)^* \\ & \mathbf{cl}(g) & \mapsto & (\mathbf{cl}(g), \mathbf{cl}(g), \cdots, \mathbf{cl}(g)) \end{array}$$

onde $\mathbf{cl}(g)$ é a classe de equivalência de g no anel apropriado.

Como $\left(\frac{\mathbb{F}_q[X]}{(v_j)}\right)^*$ possui um gerador u_j e $\#\left(\left(\frac{\mathbb{F}_q[X]}{(v_1)}\right)^*\right) = 2^{e_j} r_j$, tem-se que $(u_j^l)^r \equiv u_j^{lr} \equiv 1 \mod (v_j)$ se, e somente se, $2^{e_j} r_j | lr$. Observe que $2^{e_j} r_j | lr$ se, e somente se $\frac{2^{e_j} r_j}{(r, r_j)} | l \frac{r}{(r, r_j)}$ onde (r, r_j) denota o maior divisor comum de r e r_j, e além disso $(r, r_j) = (2^e r, r_j)$ pois r_j é ímpar. Como $1 \leq l \leq 2^{e_j} r_j$, há exatamente (r, r_j) l's para os quais isto ocorre, sendo

$$l = \frac{2^{e_j} r_j}{(r, r_j)}, \quad 2\frac{2^{e_j} r_j}{(r, r_j)}, \quad \cdots, \quad (r, r_j)\frac{2^{e_j} r_j}{(r, r_j)} = 2^{e_j} r_j.$$

Consequentemente o número de soluções da congruência $g^r \equiv 1 \mod (v_j)$ com $\mathbf{gr}(g) \leq \mathbf{gr}(v_j)$ é igual a (r, r_j). Logo, o número de soluções da congruência $g^r \equiv 1 \mod (f)$ com $\mathbf{gr}(g) \leq \mathbf{gr}(f)$ é igual a $(r, r_1)(r, r_2) \cdots (r, r_s) \leq r_1 r_2 \cdots r_s$.

Observe agora que $g^{2^i r} \equiv -1 \mod (f)$ com $0 \leq i < e$ se, e somente se, $g^{2^i r} \equiv -1 \mod (v_j)$ para todo $j = 1, 2, \cdots, s$. Se u_j é um gerador de $\left(\frac{\mathbb{F}_q[X]}{(v_s)}\right)^*$, então $(u^l)^{2^i r} \equiv u^{2^i l r} \equiv -1 \mod v_j$ se, e somente se, $2^{e_j} r_j | 2^{i+1} lq$ mas $2^{e_j} r_j \nmid 2^i lq$, ou ainda como r e r_j são ímpares deve-se ter $2^{e_j} | 2^{i+1} l$ e $2^{e_j} \nmid 2^i l$. Caso $e_j \leq i$, então o número de soluções para

$g^{2^i} r \equiv -1 \mod (v_j)$ é zero.

Suponha agora que $i < e_j$, assim

$$2^{e_j - i} \nmid l \qquad \text{e} \qquad 2^{e_j - i} \frac{r_j}{(r, r_j)} | 2l \frac{r}{(r, r_j)}.$$

Como $1 \leq l \leq 2^{e_j} r_j$, existem exatamente $2^i (r, r_j)$ l's para os quais isso ocorre, sendo

$$l = 2^{e_j - i - 1} \frac{r_j}{(r, r_j)}, \quad 3 \cdot 2^{e_j - i - 1} \frac{r_j}{(r, r_j)}, \quad \cdots, \quad (2^{i+1}(r, r_j) - 1) \cdot 2^{e_j - i - 1} \frac{r_j}{(r, r_j)},$$

isto é, os múltiplos ímpares de $2^{e_j - i - 1} \frac{r_j}{(r, r_j)}$ entre 1 e $2^{e_j} q_j$. Assim o número de soluções da congruência $g^{2^i r} \equiv -1 \mod v_j$ é igual a $2^i (r, r_j)$ quando $i < e_j$ e 0 caso contrário. Sendo $e_{min} = \min\{e_1, e_2, \cdots, e_s\}$, tem-se que o número de soluções da congruência $g^{2^i r} \equiv -1 \mod (f)$ onde $i < e_{min}$ é igual a $2^i (r, r_1) 2^i (r, r_2) \cdots 2^i (r, r_s) \leq 2^{si} r_1 r_2 \cdots r_s$.

Por fim concluímos que o número de bases g com $\mathbf{gr}(g) < \mathbf{gr}(f)$ para os quais f é pseudo-primo é menor ou igual a

$$r_1 r_2 \cdots r_s (1 + 1 + 2^r + 2^{2r} + \cdots + 2^{(e_{min} - 1)r}).$$

Assim, sua proporção entre os polinômios de grau menor que o grau de f é menor que

$$\begin{aligned}
\frac{r_1 r_2 \cdots r_s (1 + 1 + 2^r + 2^{2r} + \cdots + 2^{(e_{min} - 1)r})}{2^{e_1} r_1 2^{e_2} r_2 \cdots 2^{e_s} r_s} &= 2^{-(e_1 + \cdots + e_s)} \left(1 + \frac{2^{s e_{min}} - 1}{2^s - 1}\right) \\
&\leq 2^{-s e_{min}} \left(1 + \frac{2^{s e_{min}}}{2^s - 1} - \frac{1}{2^s - 1}\right) \\
&= \frac{1}{2^s - 1} + \frac{2^s - 2}{2^{s e_{min}} (2^s - 1)} \\
&\leq \frac{1}{2^s - 1} + \frac{2^s - 2}{2^s (2^s - 1)} \\
&= \frac{2^s + 2^s - 2}{2^s (2^s - 1)} \\
&= \frac{1}{2^{s-1}} \\
&\leq \frac{1}{2}
\end{aligned}$$

□

Seja p um número primo e considere $f \in \mathbb{Z}_p[X]$ um polinômio de grau k e $g \in \mathbb{Z}_p[X]/(f)$ escolhidos aleatoriamente. Se f satisfaz as condições (i) e (ii) da definição 5.10 para a base g, então a probabilidade de f ser escolhido como um polinômio não-irredutível é menor ou igual a $1/2$. Se são escolhidos aleatoriamente $g_1, g_2, \cdots, g_n \in \mathbb{Z}_p[X]/(f)$ e f satisfaz

a mesma condição para todas estas bases, então a probabilidade de f ter sido escolhido não-irredutível é menor ou igual a $= 1/2^n$. É claro que se a lista $g_1, g_2, \cdots, g_n$ representa metade dos elementos de $\mathbb{Z}_p[X]/(f)$, então f é irredutível.

Deste resultado obtêm-se um método probabilístico para encontrar polinômios irredutíveis em $\mathbb{Z}_p[X]$ e com isso trabalhar com corpos finitos $\mathbb{F}_q$. Tal algoritmo é apresentado abaixo.

Algoritmo 5.12. *Polinômios Irredutíveis em $\mathbb{Z}_p$*

Entrada*: Dois inteiros positivos n e k e um inteiro primo p*

Saída*: Um polinômio $f \in \mathbb{Z}_p[X]$ de grau k escolhido com chances $1-1/2^n$ de ser irredutível*

---

1. Coloque $q \leftarrow p^k - 1$ e $e \leftarrow 0$;

2. Enquanto q é par coloque $q \leftarrow q/2$ e $e \leftarrow e + 1$;

3. escolha aleatoriamente um polinômio $f \in \mathbb{Z}_p[X]$ de grau k;

4. coloque $i \leftarrow 0$;

5. se $i = n$ pare;

6. escolha aleatoriamente $g \in \mathbb{Z}_p[X]/(f)$ com $g \neq 0$;

7. coloque $g \leftarrow g^q$;

8. se $g = \pm 1$, coloque $i \leftarrow i + 1$ e volte ao passo 5;

9. coloque $j \leftarrow 0$;

10. coloque $g \leftarrow g^2$ e $j \leftarrow j + 1$;

11. se $g \neq -1$ coloque $i \leftarrow i + 1$ e volte ao passo 5;

12. se $j < e$ volte ao passo 10;

13. volte ao passo 3.

Observe que no algoritmo 5.12, $\mathbb{Z}_p$ pode substituído pelo corpo $\mathbb{F}_q$ sem qualquer outra alteração, entretanto esta substituição é desnecessária já que $\mathbb{Z}_p[X]/(f)$ com f irredutível de grau k e $\mathbb{F}_{p^k}$ são isomorfos.

Supondo que um polinômio irredutível é encontrado na primeira vez que o passo 3 deste algoritmo é executado, para verificar que este é realmente irredutível, este algoritmo executa n vezes o passo 7 e ne vezes o passo 10. Respectivamente o custo do passo 7 e do passo 10 são $O(\log_2 q)$ vezes uma multiplicação em $\mathbb{Z}_p[X]/(f)$ e uma vezes a multiplicação neste mesmo anel. O valor de n é dado como entrada do algoritmo, porem o valor de e dependerá de p e k podendo ser qualquer valor entre 0 e $\log_2(p^k - 1)$. Uma rápida contagem dos múltiplos de 2, 4 , 8, assim por diante entre 0 e $\log_2(p^k - 1)$ concluirá que o valor esperado para e não ultrapassa 2.

Assim, o custo para o algoritmo5.12 verificar se um dado polinômio é irredutível é $O(n \log_2 q + 2n \log_2(p^k - 1))$. O custo real deste algoritmo só pode ser calculado conhecendo a porcentagem dos polinômios irredutíveis dentre todos os polinômios de grau k, porem encontrar a expressão que dá a quantidade de polinômios irredutíveis de grau k é um

tanto complicada quando maior o valor de k. Ao invés de calcular esta expressão é apresentado no próximo capítulo os gráfico do uso deste algoritmo com relação a quantidade de tentativas necessárias para encontrar tais polinômios.

6 RESULTADOS PRÁTICOS

Neste capítulo é apresentado uma comparação dos resultados obtidos pela execução dos algoritmos expostos nos capítulos anteriores com os resultados teóricos descritos anteriormente. Os algoritmos foram implementados em linguagem C/C++ e executados utilizando o compilador GCC 4.6 (GNU Compiler Collection) para o sistema operacional linux. A versão linux utilizada foi o Ubuntu 12.04 e a máquina um notebook Dell Inspiron-N4050 com processador Core i5 e 4Gb de memória ram.

Durante a execução dos algoritmos verificou-se a utilização de apenas um dentre os 4 núcleos do processador, com isso acredita-se que processos paralelos do próprio sistema operacional e de outros programas em execução não foram relevantes na alteração do tempo de execução destes algoritmos.

Para trabalhar com a aritmética modular $\mathbb{Z}_n$ foram utilizados as funções da biblioteca bn.h do pacote de criptografia OpenSSL [13]. Os algoritmos para trabalhar com a aritmética dos anéis $\mathbb{Z}_n[X]$ e dos corpos finitos $\mathbb{F}_q$ foram implementados.

6.1 POLINÔMIOS

Uma ideia para comportamento da distribuição dos polinômios irredutíveis de grau fixo k sobre $\mathbb{Z}_p[X]$, onde p é primo, foi obtida executando o algoritmo 5.12 para $p = 3, 5, 7$ e 11 com k variando entre 10 a 50. Para cada valor de k o algoritmo foi executado 5 vezes obtendo uma média de tentativas para encontrar um polinômio irredutível apresentada na tabela 6.1.

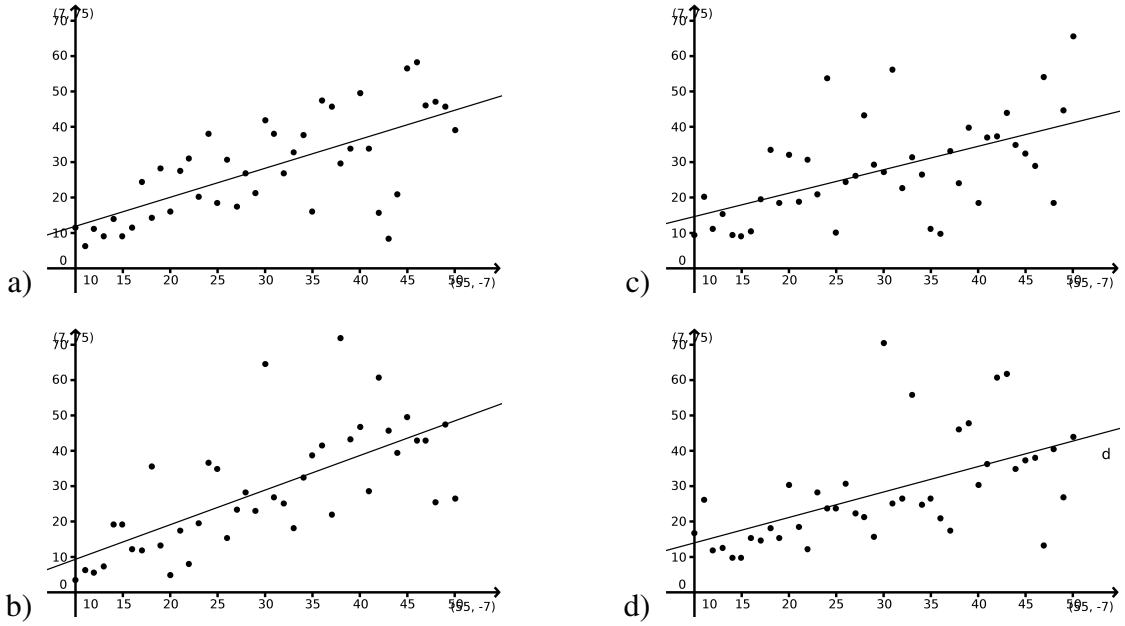
Tabela 6.1: Número médio de tentativas para encontrar um polinômio de grau k sobre $\mathbb{Z}_p[X]$

$k\backslash p$	3	5	7	11		$k\backslash p$	3	5	7	11		$k\backslash p$	3	5	7	11
10	11,6	3,4	9,4	16,8		24	38	36,6	53,6	23,6		38	29,8	72	24	46
11	6,2	6,4	20,2	26		25	18,6	35	10,2	23,6		39	34	43,2	39,8	47,8
12	11,2	5,6	11	12		26	30,8	15,2	24,4	30,8		40	49,6	46,6	18,6	30,4
13	9,2	7,2	15,4	12,6		27	17,6	23,4	26,2	22,4		41	34	28,6	36,8	36,2
14	13,8	19,2	9,4	9,8		28	26,8	28,2	43,2	21,2		42	15,8	60,8	37,2	60,6
15	9,2	19	9	9,6		29	21,2	23	29,2	15,6		43	8,4	45,6	44	61,8
16	11,4	12,2	10,4	15,2		30	41,8	64,6	27,2	70,6		44	21	39,4	35	35
17	24,4	11,8	19,2	14,8		31	38	26,8	56,2	25,2		45	56,6	49,4	32,6	37,4
18	14,4	35,4	33,4	18,2		32	27	25	22,6	26,6		46	58,4	43	29	38
19	28,2	13,4	18,4	15,4		33	32,8	18	31,4	55,8		47	46,2	42,8	54,2	13,2
20	16,2	4,8	32,2	30,4		34	37,6	32,6	26,4	24,8		48	47,2	25,6	18,6	40,6
21	27,4	17,4	18,8	18,4		35	16,2	38,6	11	26,6		49	45,6	47,6	44,8	27
22	31,2	8	30,8	12,2		36	47,4	41,4	9,6	20,8		50	39,2	26,6	65,6	43,8
23	20,2	19,4	20,8	28,4		37	45,8	21,8	33	17,4						

Observa-se que a quantidade de tentativas necessárias para encontrar um po-

linômio irredutíveis de grau k aumenta de acordo com acréscimos em k, esta tabela indica que a razão dentro o número polinômios de k irredutíveis pelo número de polinômios de mesmo grau em $\mathbb{Z}_p[X]$ diminui a medida que k aumenta. Isto pode ser melhor observado pelo diagrama de dispersão e o gráfico da reta de regressão linear dos dados da tabela 6.1 na figura de 6.1.

Figura 6.1: Diagramas de dispersão e retas de regressão linear para os dados da tabela 6.1



(a) Diagrama de dispersão e reta de regressão linear para $p = 3$ (b) Diagrama de dispersão e reta de regressão linear para $p = 5$ (c) Diagrama de dispersão e reta de regressão linear para $p = 7$ (d) Diagrama de dispersão e reta de regressão linear para $p = 11$

Não há necessidade, na criptografia de curvas elípticas, de utilizar representações diferentes para um mesmo corpo finito, a menos de isomorfismo. Assim, a probabilidade cada vez menor de encontrar polinômios irredutíveis não inviabiliza o algoritmo 5.12, pois não há necessidade de encontrar mais de um polinômio dados $\mathbb{Z}_p$ e k.

Neste trabalho utilizou-se o algoritmo 5.12 para encontrar um polinômio irredutível sobre $\mathbb{Z}_p$, com p variando entre os número primos de 3 a 100 com grau suficiente para trabalhar em chaves de criptografia de 64, 128 e 256 bits. Para um número primo p, o grau do polinômio necessário para codificar uma chave de N bits é dado por $\left\lfloor \frac{N}{\log_2 p} + 1 \right\rfloor$. A tabela 6.2 apresenta o tempo gasto necessário para encontrar tais polinômios.

Observando a tabela 6.2 pode-se ver que determinar polinômios irredutíveis pelo algoritmo 5.12 é viável para estes tamanhos de chaves.

Tabela 6.2: Tempo necessário para obter um polinômio irredutível em $\mathbb{Z}_p$ dado o tamanho da chave

número primo p	64 bits	128 bits	256 bits
3	0:00:23.158	0:27:42.237	1:01:46.318
5	0:00:23.732	0:02:02.452	3:35:15.266
7	0:00:06.174	0:03:28.358	0:51:13.965
11	0:00:01.558	0:00:15.855	0:08:54.170
13	0:00:02.397	0:01:51.496	0:26:40.873
17	0:00:03.318	0:00:13.201	0:04:54.235
19	0:00:04.792	0:00:47.391	0:10:46.949
23	0:00:02.988	0:00:04.770	0:05:43.597
29	0:00:01.405	0:00:14.060	0:06:01.203
31	0:00:01.924	0:00:14.305	0:04:04.507
37	0:00:01.827	0:00:50.147	0:04:49.393
41	0:00:01.509	0:00:06.682	0:13:03.302
43	0:00:01.065	0:00:10.819	0:01:58.139
47	0:00:01.231	0:00:08.953	0:06:51.497
53	0:00:01.309	0:00:07.922	0:03:23.532
59	0:00:01.030	0:00:06.878	0:01:05.940
61	0:00:00.588	0:00:07.757	0:05:04.148
67	0:00:01.471	0:00:15.535	0:03:47.608
71	0:00:02.571	0:00:08.425	0:00:32.773
73	0:00:00.681	0:00:09.098	0:02:00.181
79	0:00:00.966	0:00:06.168	0:00:33.204
83	0:00:00.769	0:00:23.887	0:01:51.387
89	0:00:01.004	0:00:08.289	0:03:13.109
97	0:00:01.377	0:00:11.275	0:01:32.958

6.2 MULTIPLICAÇÃO DE UM PONTO POR UM INTEIRO

Para utilizar usando o algoritmo 5.5 é necessário escrever o inteiro n como uma combinação das potências de τ como mostra a equação (5.4), onde $\tau^2 - t\tau + q = 0$. Para isso executamos o algoritmo 5.4 algumas vezes escolhendo de modo aleatório q dentre os primos e potências de primos entre 0 e 100, t satisfazendo $|t| \leq 2\sqrt{q}$ e n necessário para codificar uma mensagem com 128 bits. Em todos os exemplos executados verificou-se que o valor de l da equação (5.4) não ultrapassa $2(\log_q n + 1)$.

Para comparar a multiplicação de pontos de curvas elípticas por inteiros usando os algoritmos 2.6 e 5.5 foi escolhida uma curvas elíptica de modo aleatório para cada corpo $\mathbb{Z}_p$ com p variando dentre os primos de 3 a 100. Em cada curva elíptica foram escolhidos de modo aleatório uma lista de 10 inteiros n e 10 pontos P sobre a curva.

As figuras 6.2, 6.3 e 6.4 apresentam os gráficos percentuais do tempo médio gasto pelos algoritmos 5.5 e 2.6 para os tamanhos de chaves 64, 128 e 256 bits juntamento com os gráficos percentuais para o custo assintótico destes algoritmos.

Figura 6.2: Tempo percentual gasto para uma chave de tamanho de 64 bits

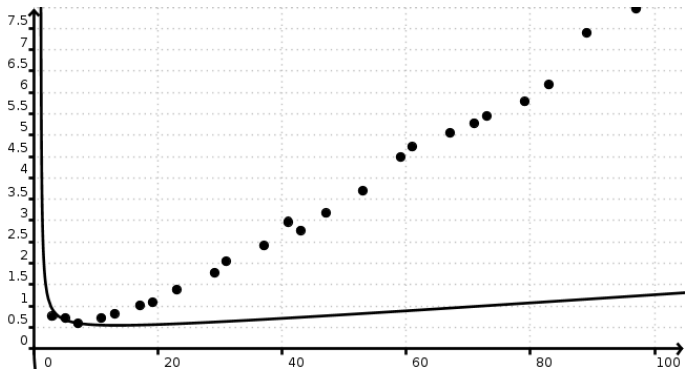


Diagrama de dispersão do tempo percentual gasto pelo algoritmo 5.5 com relação ao algoritmo 2.6 e gráfico da previsão do tempo percentual para uma chave de tamanho de 64 bits

Figura 6.3: Tempo percentual gasto para uma chave de tamanho de 128 bits

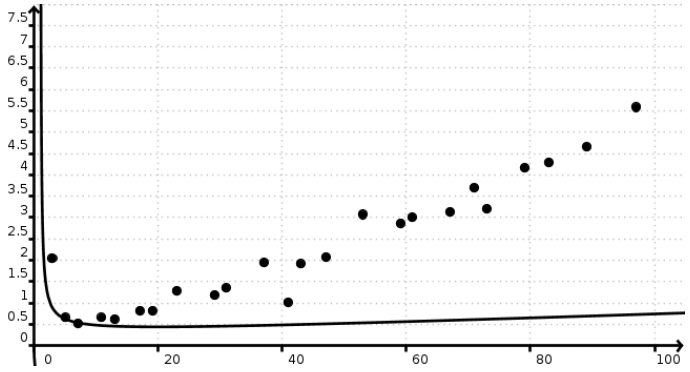


Diagrama de dispersão do tempo percentual gasto pelo algoritmo 5.5 com relação ao algoritmo 2.6 e gráfico da previsão do tempo percentual para uma chave de tamanho de 128 bits

Como previsto pelas curvas contínuas, o algoritmo 5.5 apresenta vantagem em tempo de execução com relação ao algoritmo 2.6, entretanto o comportamento obtido pela execução dos algoritmos diferem em alguns aspectos dos resultados teóricos. Isto se deve a substituição feita nas expressões (5.8) e (5.9) de $\mathcal{M}$ e $\mathcal{I}$ por N^2 e N^3 respectivamente. Um outro fator para a diferença entre os resultados teóricos e os resultados práticos talvez seja a utilização dos algoritmos da biblioteca bn.h do pacote OpenSSL, que já a alguns anos veem sendo melhorados, juntamente com os algoritmos aqui implementados.

Figura 6.4: Tempo percentual gasto para uma chave de tamanho de 256 bits

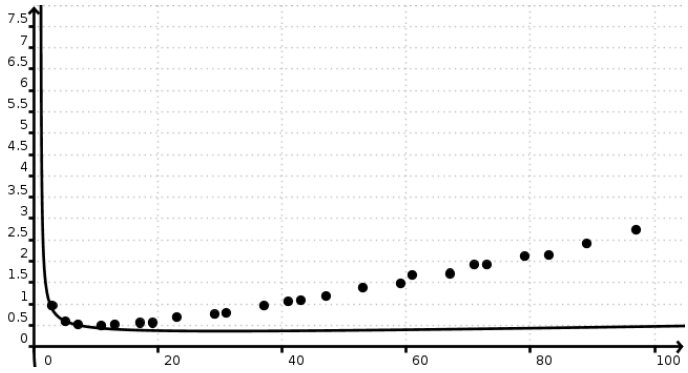


Diagrama de dispersão do tempo percentual gasto pelo algoritmo 5.5 com relação ao algoritmo 2.6 e gráfico da previsão do tempo percentual para uma chave de tamanho de 256 bits

7 CONCLUSÃO

Neste trabalho é estudado algumas propriedades de curvas elípticas definidas sobre corpos finitos no uso da criptografia. Apresenta-se uma definição de um algoritmo de criptografia de chave pública e o algoritmo de ElGamal para grupos. É dada uma breve apresentação para o problema do Logaritmo Discretos, sua relação com o algoritmo de ElGamal, e um possível método para resolvê-lo quando conhecido a fatoração da ordem do grupo utilizado.

Também é apresentado brevemente alguns resultados sobre extensões, característica e cardinalidade de corpos finitos e como podem ser representados por meio de anéis quocientes para ser operados eficientemente na prática.

Sobre curvas elípticas, apresenta-se sua definição, o algoritmo de adição de pontos, resultados envolvendo simplificações sobre certos corpos, a ordem e a estrutura do grupo de pontos. Foi apresentada a utilização do algoritmo de ElGamal com o grupo de pontos de uma curva elíptica, o problema do logaritmo discreto sobre curvas elípticas (ECDLP) e o algoritmo MOV para este problema específico. Nesta altura, é indicado os cuidados necessários a serem tomados ao escolher uma curva elíptica para criptografia.

Fez-se referências aos endomorfismos de curvas elípticas, em especial, o endomorfismo de Frobenius. A partir das propriedades deste endomorfismo conseguiu-se desenvolver um algoritmo para a multiplicação de pontos de curvas elípticas por inteiros que, para algumas curvas elípticas definidas em extensões de corpos finitos pequenos, é mais eficiente do que o algoritmo tradicional para a mesma operação, embora não seja tão eficiente quando a adição feitas em coordenadas projetivas, de Jacob e de Edward. Apresentou-se ainda as curvas de Koblitz que possui uma ideia semelhante em corpos de característica 2.

Ainda para a mesma família de curvas é proposto um outro método para determinar a ordem do grupo de pontos de maneira rápida por meio de uma recorrência linear, algo importante para a determinação da segurança de uma curvas elíptica para a criptografia.

Por fim, foi descrito um algoritmo capaz de determinar, com uma probabilidade controlada, a irredutibilidade de polinômios sobre corpos finitos. Quando implementados em linguagem C/C++, os resultado observados na prática confirmaram as previsões teóricas.

Muitos das propriedades de curvas elípticas não são unicamente restritas à criptografia. Por exemplo a conjectura de Taniyama-Shimura relaciona o estudo das curvas elípticas com formas modulares, tal conjectura, demonstrada por Andrews Willes em 1995, tem como corolário o resultado do último teorema de Fermat. Antes disso casos particulares deste teorema já vinham sendo demonstrados utilizando curvas elípticas e a teoria dos números algébricos.

O resultado para polinômios irredutíveis é aqui obtido considerando uma generalização do conceito de elementos primo e irredutível no conjunto dos números inteiros para domínios euclidianos, especificamente em $\mathbb{F}_q[X]$. Os diagramas do capítulo 7 sugere um de-

crescimento da ordem de $1/k$ para a porcentagem de polinômios irredutíveis de grau k quando k aumenta, será porem que a distribuição de polinômios irredutíveis de grau k dentre o conjunto de polinômios de mesmo grau em $\mathbb{F}_q[X]$ assemelha-se com a distribuição dos números primos? Isto levanta a questão sobre até que ponto os resultados de números primos podem ser generalizados para outros domínios de ideais principais.

REFERÊNCIAS

[1] BOREVICH, Z. I., AND SHAFAREVICH, I. R. Number theory. Academic Press, New York, NY, 1966.

[2] CARELLA, N. A. Topic In Elliptic Curves Over Finite Fields: The Groups of Points. *ArXiv e-prints* (2011).

[3] COHEN, H. *A course in computational algebraic number theory*. Springer-Verlag, 1993.

[4] COHEN, H., AND FREY, G. *Handbook of elliptic and hyperelliptic curve cryptography*. Chapman & Hall/CRC, Abingdon, 2005.

[5] COUTINHO, S. *Número Inteiros e Criptografia RSA*. IMPA, Rio de Janeiro, RJ, 2005.

[6] DE ARAÚJO NETO, A. C. Um algoritmo de criptografia de chave pública semanticamente seguro baseado em curvas elípticas. Instituto de Informática, Universidade Federal do Rio Grande do Sul, 2006.

[7] ELGAMAL, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory 31*, 4 (1985), 469–472.

[8] HOFFSTEIN, J., PIPHER, J. C., AND SILVERMAN, J. H. *An Introduction to Mathematical Cryptography*. Springer-Verlag, 2008.

[9] LANG, S. *Algebra*. Addison-Wesley, Reading, MA, 1970.

[10] LEMOS, M. Criptografia, números primos e algoritmos. In *17º Colóquio Brasileiro de Matemática* (IMPA/CNPq, julho 1989).

[11] MENEZES, A., VANSTONE, S., AND OKAMOTO, T. Reducing elliptic curve logarithms to logarithms in a finite field. ACM, pp. 80–89.

[12] MILLER, V. S. The Weil pairing, and its efficient calculation. *Journal of Cryptology 17*, 4 (2004), 235–261.

[13] OPENSSL: CRIPTOGRAPHY AND SSL/TLS TOOLKIT. <http://www.openssl.org>, Acesso em: 10 de março de 2012.

[14] ROMAN, S. *Field Theory*, vol. 158. Springer-Verlag, 2006.

[15] SCHOOF, R. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux 7*, 1 (1995), 219–254.

[16] SHANKS, D. Class number, a theory of factorization, and genera. In *1969 Number Theory Institute (Proc. Sympos. Pure Math., Vol. XX, State Univ. New York, Stony Brook, N.Y., 1969)*. Providence, R.I., 1971, pp. 415–440.

[17] SHIKATA, J., ZHENG, Y., SUZUKI, J., AND IMAI, H. Realizing the menezes-okamoto-vanstone (MOV) reduction efficiently for ordinary elliptic curves. *TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems* (2000).

[18] SILVERMAN, J. H. *Arithmetic of Elliptic Curves*. Springer-Verlag, 1986.

[19] SILVERMAN, J. H., AND TATE, J. *Rational Points on Elliptic Curves*. Springer-Verlag, 1992.

[20] STOER, J., AND BULIRSCH, R. *Introduction to Numerical Analysis*, 3 ed. Springer, New York, NY, 2002.

[21] WASHINGTON, L. C. *Elliptic Curves, Number Theory and Cryptography*. Discrete Mathematics and Its Applications. Chapman & Hall/CRC, 2003.

[22] WILES, A. Modular elliptic curves and fermat's last theorem. *The Annals of Mathematics 141*, 3 (1995), 443–551.

Índice Remissivo

j-invariante, 35

adição de pontos, 37
algoritmo de criptografia, 18
algoritmo de ElGamal, 21
algoritmo MOV, 49
anel, 27
    com divisor de zero, 27
    com identidade, 27
    comutativo, 27
    de divisão, 28
    sem divisor de zero, 27
anel de polinômios, 32
anel quociente, 30
ASCII - American Standard Code for Information Interchange, 17

característica de um corpo, 31
CM - multiplicação complexa, 48
coordenadas
    de Edwards, 46
    jacobianas, 45
    projetivas, 43
corpo, 28
    finito, 33
corpo quadrático, 28
criptografia de chave pública, 19
criptografia de curvas elípticas, 47
curva elíptica, 35
    singular, 35
    super-singular, 50
curvas de Koblitz, 56
curvas elíptica
    não-singular, 35
custo assintótico de execução, 22
custo de execução, 22
custo exponencial de execução, 24
custo polinomial de execução, 24
custo subexponencial de execução, 24

DHP - Problema de Diffie-Helman, 23
discriminante, 35
DLP - Problema do Logaritmo Discreto, 23
domínio de ideais principais, 30
domínio de integridade, 28

ECDLP - porblema do logaritmo discreto sobre curvas elípticas, 48
emparelhamento de Weil, 49
endomorfismo, 39
endomorfismo de frobenius, 41
equação de Weierstrass, 35
extensão de corpos, 28

grau de extensão de corpos, 33
grau de polinômio, 32
grupo, 20
    abeliano, 20
    aditivo, 20
grupo de pontos de uma curvas elíptica, 38

homomorfismo de anéis, 29

ideal, 29
    gerado, 30
    maximal, 31
    principal, 30
isomorfismo de anéis, 29

ordem, 40

plano projetivo, 43
polinômio, 31
    irredutível, 33
ponto de singularidade, 35
pseudo-primo, 58, 59

subanel, 28
subcorpo, 28

termos de Tate, 35
torção, 48