Protocolos de acordo de chaves baseados em emparelhamentos, para dispositivos móveis

Cleber Morio Okida

Dissertação apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Mestre em Ciências

Programa: Ciência da Computação

Orientador: Prof. Dr. Routo Terada

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da FAPESP – Processo Nº 08/50412-5

São Paulo, agosto de 2011

Protocolos de acordo de chaves baseados em emparelhamentos, para dispositivos móveis

Esta dissertação trata-se da versão original
do aluno Cleber Morio Okida.

Resumo

A telemedicina é um ramo emergente da medicina que se aproveita de tecnologias de telecomunicações para tratar pacientes em locais remotos.

Com o advento dos telefones celulares e outras tecnologias de telecomunicações móveis, a telemedicina tem confiado cada vez mais nessas tecnologias para oferecer uma assistência à saúde de melhor qualidade à população.

Porém, é necessário um sistema provedor de privacidade e segurança da informação que trafega entre esses dispositivos móveis conectados com um servidor de banco de dados.

Além disso, muitas dessas atividades são realizadas usando dispositivos móveis, limitados em termos de disponibilidade de energia, memória e poder computacional. A criptografia de curva elíptica (CCE) oferece o mais alto nível de segurança por tamanho da chave (em bits) comparado com qualquer outro sistema de chave pública conhecido.

Os acordos de chaves têm a finalidade de permitir que dois usuários negociem uma chave com segurança, a qual pode então ser usada para a subsequente criptografia das mensagens.

A criptografia sem certificados combina o melhor da criptografia tradicional de chave pública e com o paradigma baseado em identidade, porque não há necessidade de obter e verificar os certificados, e as chaves particulares não estão sob custódia do centro de geração de chaves (KGC).

Neste trabalho, nós aplicamos a CCE e os emparelhamentos bilineares sobre curvas elípticas para criar um protocolo de acordo de chave autenticado sem certificado. Com a finalidade de obter uma chave de sessão para uso posterior em um algoritmo simétrico para transferir os dados dos prontuários médicos.

O desenvolvimento da biblioteca de criptografia em Java que permite usar curva elíptica Barreto-Naehrig sobre um corpo primo arbitrário e grau de mergulho $k = 12$, forneceu um elevado nível de segurança com o parâmetro de segurança menor que o RSA para comunicação criptografada entre o dispositivo móvel e o servidor de banco de dados.

Palavras-chave: acordo de chaves, sem certificados, emparelhamento bilinear, criptografia de chave pública, dispositivos móveis.

Abstract

Telemedicine is an emerging branch of medicine that takes advantage of telecommunication technologies to treat patients in remote locations.

With the advent of cell phones and other mobile technology, telemedicine has increasingly relied on these technologies to provide health care of better quality to the population.

However, there must be a system providing privacy and security of information that goes between these mobile devices connected to a database server.

In addition, many of these activities are performed using small, mobile devices, which are severely limited in terms of available energy, memory, and computational power. The Elliptic Curve Cryptography (ECC) delivers the highest security strength per bit of key in any known public key system.

The key agreements are intended to enable two users to securely exchange a key that can then be used for subsequent encryption of messages.

Certificateless cryptography (CL-PKC) combines the best of traditional public-key cryptography and the ID-based (ID-PKC) paradigm. There is no need to get and verify certificates, and the secret keys are not under the key generation center (KGC)'s escrow.

In this work, we apply the ECC and the bilinear pairings on elliptic curves to create a certificateless authenticated key agreement (CL-AKA) protocol in order to obtain a session key for later use of symmetric algorithm for transferring data of medical records.

The development of a generic cryptographic library in Java that allows us the use the Barreto-Naehrig elliptic curve over an arbitrary prime field and embedding degree $k = 12$, provided a high security level, with the security parameter smaller than the RSA algorithm parameter, through encrypted communication between a mobile device with a database server.

Keywords: key agreement, certificateless, bilinear pairing, public key cryptography, mobile devices.

Sumário

Lista de Abreviaturas ix

Lista de Símbolos xi

Lista de Figuras xiii

Lista de Tabelas xv

1 Introdução 1

1.1 Motivação . . . 2
1.2 Objetivos . . . 3
1.3 Organização do Trabalho . . . 4

2 Fundamentos de Criptografia de Chave Pública 5

2.1 Modelos de criptografia de chave pública . . . 7
2.2 Modelo criptografia baseado em identidade . . . 8
2.3 Criptografia sem certificado . . . 8
2.4 Protocolos de acordo de chaves . . . 9
2.5 Modelo do oráculo aleatório e suas implicações . . . 9
2.6 Problemas de interesse . . . 11
2.7 Trabalhos correlatos . . . 13

3 Conceitos preliminares 15

3.1 Curvas Elípticas . . . 15
3.2 Aritmética de curvas elípticas . . . 15
3.3 Torções em curvas elípticas . . . 18
3.3.1 Torções de alto grau . . . 18
3.4 Lei de grupos de curvas elípticas . . . 19
3.5 Coordenadas Projetivas . . . 20
3.5.1 Operações de pontos em coordenadas projetivas . . . 21
3.6 Estruturas algébricas de curvas elípticas . . . 23
3.6.1 Ordem da curva elíptica . . . 24
3.6.2 Tipos de curvas . . . 25
3.6.3 Grupo multiplicativo . . . 25
3.7 Mais alguns conceitos . . . 25
3.7.1 Multiplicação complexa . . . 26

3.7.2 Raízes n-ésimas da unidade . . . 26
3.7.3 Polinômios ciclotômicos . . . 26
3.8 Criptografia de Curvas Elípticas . . . 27
3.9 Resumo . . . 28

4 Emparelhamento bilinear **29**
4.1 Emparelhamento de Weil . . . 31
4.1.1 Emparelhamento de Tate . . . 32
4.2 Exponenciações . . . 34
4.2.1 Propriedades do emparelhamento de Tate . . . 36
4.3 Algoritmo de Miller . . . 38
4.4 Emparelhamento de Ate . . . 41
4.5 Curvas elípticas ordinárias de emparelhamento amigável . . . 44
4.6 Curvas BN (Barreto-Naehrig) . . . 45
4.7 Resumo . . . 46

5 Protocolos implementados **47**
5.1 Modelo de segurança para esquemas de acordo de chave sem certificados . . . 48
5.2 Arquitetura dos Protocolos . . . 49
5.3 Descrição do protocolo de Lippold-Boyd-González . . . 50
5.3.1 Instalação . . . 50
5.3.2 Troca de mensagens . . . 51
5.3.3 Cálculo das chaves . . . 51
5.3.4 Considerações de eficiência . . . 52
5.4 Descrição do protocolo de Goya-Okida-Terada . . . 53
5.4.1 Instalação . . . 53
5.4.2 Chaves do usuário . . . 53
5.4.3 Troca de mensagens . . . 53
5.4.4 Cálculo das chaves . . . 54
5.4.5 Considerações de eficiência . . . 54
5.5 Protocolo proposto . . . 55
5.5.1 Instalação . . . 55
5.5.2 Chaves do usuário . . . 56
5.5.3 Troca de mensagens . . . 56
5.5.4 Cálculo das chaves . . . 56
5.5.5 Considerações de eficiência . . . 57
5.6 Resumo . . . 57

6 Implementação **59**
6.1 Operações de precisão arbitrária . . . 59
6.1.1 Método de multiplicação de Karatsuba . . . 60
6.2 Operações em corpos finitos . . . 61
6.3 Operações em curvas elípticas de característica prima . . . 62
6.4 Curvas para calcular emparelhamentos . . . 62

6.4.1 Eficiência relativa dos parâmetros das curvas para emparelhamento amigável . 64
6.5 Implementação de emparelhamentos assimétricos 65
6.6 Implementação do Emparelhamento Ate 66
6.7 Hash em $\mathbb{G}_1$ e em $\mathbb{G}_2$ 67
6.8 Estudo de caso – Projeto Borboleta 68
6.9 Resumo 69

7 Experimentos **71**

8 Conclusões **75**
8.1 Considerações Finais 75
8.2 Sugestões para Pesquisas Futuras 75

A Conceitos Básicos e Propriedades **77**
A.1 Grupo 77
A.1.1 Homomorfismo 78
A.1.2 Relação de equivalência 78
A.2 $\mathbb{Z}$, $\mathbb{Z}_n$ e $\mathbb{Z}_n^*$ 78
A.3 Corpo 79
A.3.1 Corpo finito 79
A.3.2 Corpo primo 79
A.3.3 Corpo binário 79
A.3.4 Representação de corpos finitos $\mathbb{F}_{p^2}$ 80
A.3.5 Simetria rotacional 80
A.3.6 Simetria rotacional de ordem n 81
A.3.7 Raizes n-ésimas da unidade 81
A.3.8 Operações básicas com números complexos 82
A.4 Curiosidades 83

B Divisores **85**
B.1 Uma introdução intuitiva para divisores 85

Referências **93**

Lista de Abreviaturas

AKA	Esquema de acordo de chaves autenticado (*Authenticate Key Agreement*)
AKE	Esquema de estabelecimento de chaves autenticado (*Authenticate Key Estabilishment*)
BN	Curva Barreto-Naehrig
CDHP	Problema Diffie-Hellman Computacional
CL	Sem certificado (*Certificateless*)
CM	Algoritmo de multiplicação complexa (*Complex multiplication*)
DDHP	Problema de Decisão de Diffie-Hellman
DLP	Problema do Logaritmo Discreto
CCE/ECC	Criptografia de curva elíptica (*Elliptic Curve Cryptography*)
GDHP	Problema Gap Diffie-Hellman
IBE	Cifragem baseado em identidade (*Identity Based Encryption*)
ICP/PKI	Infraestrutura de chaves públicas (*Public Key Infrastructure*)
ID	baseado em Identidade (*Identity based*)
KGC	Centro de gerador de chaves (*Key Generator Center*)
MC	Multiplicação complexa (*Complex multiplication*)
PDA	Assistente pessoal digital (*Personal digital assistants*)
PE	Prontuário eletrônico
PK	Chave Pública (*Public Key*)
PKG	Gerador de chaves particulares (*Private Key Generator*)
RSA	Algoritmo Rivest-Shamir-Adleman
SBIS	Sociedade Brasileira de Informática em Saúde
SUS	Sistema Único de Saúde
TI	Tecnologia da Informação

Lista de Símbolos

$\mathbb{G}$	Grupo (genérico)
$\mathbb{K}$	Corpo (genérico)
$\mathbb{F}$	Corpo finito
$\mathbb{F}_q$	Corpo finito com q elementos
E	Curva elíptica
$E(\mathbb{F}_q)$	Curva elíptica sobre o corpo $\mathbb{F}_q$
$\#E(\mathbb{F}_q)$	Ordem da curva elíptica sobre o corpo $\mathbb{F}_q$
Δ	Discriminante da curva elíptica
P, Q	Pontos da curva elíptica E
k	grau de mergulho
$div(u)$	Divisor u
lc	coeficiente líder de uma série de Laurent
p	um número primo
$\mathcal{O}$	Ponto no infinito
w_r	Emparelhamento de Weil sobre a curva $E[r]$
μ_n	n-ésima raíz da unidade
Φ	Polinômio ciclotômico
L	Matriz geradora do reticulado
δ	Discriminante da equação característica de Frobenius

Lista de Figuras

2.1 Exemplo de criptografia assimétrica 6

3.1 Exemplos de curvas elípticas:(a) curva não-singular (b) curva singular 16
3.2 Exemplos das operações de curvas elípticas 20
3.3 Pontos da curva elíptica sobre $\mathbb{F}_5$ 21
3.4 Exemplos do S^1 com as raízes n-ésimas da unidade 27

4.1 Exemplo da utilização de divisores em operações de curvas elípticas 35

5.1 PK-AKE + ID-AKE $\neq$ CL-AKE 50

6.1 Modelo da implementação da biblioteca criptográfica 59

B.1 Ilustração das linhas u e v na adição de pontos em uma curva elíptica. 88
B.2 Formas das linhas u e v usadas para adicionar divisores de uma curva elíptica. 89

Lista de Tabelas

3.1 Curvas elípticas e suas torções 18
3.2 Pontos nas torções de curvas elípticas 18
3.3 Pontos na curva $E/\mathbb{F}_5 : y^2 = x^3 + 1$ 20
3.4 Adição de pontos na curva $y^2 = x^3 + 1$ sobre $\mathbb{F}_5$ 20
3.5 As operações de corpo necessárias para implementar operações em curva elípticas em sistemas diferentes de coordenadas. 23
3.6 Comparação entre afins e projetivas 23
3.7 Classificação das curvas supersingulares 26

5.1 Comparação dos protocolos 55

6.1 Típos úteis de curvas ordinárias 64
6.2 Parâmetros ideais em função do nível de segurança para $\rho = 1$ usando curvas ordinárias 65
6.3 Melhores parâmetros conhecidos em função do nível de segurança usando curvas ordinárias 65

7.1 Tempo de execução para as operações do protocolo proposto 72
7.2 Tamanhos dos parâmetros de cada tipo de sistema em função do nível de segurança . 72
7.3 Tempo de execução para o protocolo proposto 73
7.4 Tempo de execução para transmissão dos dados 73

Capítulo 1

Introdução

De acordo com o dicionário Aurélio de Língua Portuguesa de Holanda Ferreira e outros (1999), Criptografia (do grego *kryptós*, "escondido", e *gráphein*, "escrita") "*é arte de escrever em cifra ou em código; conjunto de técnicas que permitem criptografar informações (como mensagens escritas, dados armazenados ou transmitidos por computador)*". Esta definição pode ser historicamente aceita, mas ela não detém a essência da criptografia moderna.

De fato, até o século XX a criptografia era uma arte. A construção de bons códigos, ou a quebra dos existentes, apoiava-se na criatividade e na habilidade pessoal. Havia pouca teoria que poderia se recorrer e não havia sequer uma noção bem definida do que constitui um bom código.

Uma rica teoria surgiu, permitindo o estudo rigoroso da criptografia como uma ciência. Como consequência, muitos objetivos foram adicionados ao meio de comunicação privada na presença do adversário, e a criptografia passou de uma arte de engenharia construída sobre uma série de técnicas heurísticas para uma disciplina científica baseada em requisitos de concepção matematicamente rigorosa, as técnicas de solução e provas de corretude (Catalano *et al.*, 2005, cap. C).

Além disso, o campo da criptografia agora abrange muito mais do que uma comunicação secreta. Por exemplo, ela lida com os problemas de autenticação de mensagens, assinaturas digitais, protocolos para troca de chaves secretas, protocolos de autenticação, leilões e eleições eletrônicas, dinheiro digital e muito mais.

Na verdade, a criptografia moderna deve estar preocupada com os problemas que podem surgir em qualquer computação distribuída que poderá estar sob ataque interno ou externo. Katz e Lindell (2007) definem a criptografia moderna como "*o estudo científico de técnicas para proteger as informações digitais, transações e computação distribuída.*"

Na RFC 4949, o termo "Criptanálise" é usado para referir-se à "*ciência matemática que lida com a análise de um sistema criptográfico, a fim de adquirir conhecimento necessário para quebrar ou contornar a proteção que o sistema é projetado para fornecer*" (Shirey, 2007).

A criptanálise é a contrapartida das noções de segurança, correspondentes às categorias de adversários. O objetivo fundamental de um criptanalista é violar, implícita ou explicitamente, uma ou várias destas noções de segurança exigidas nos algoritmos para satisfazer os critérios desses modelos de segurança (Joux, 2009).

Como tal, o criptanalista é o antagonista do criptógrafo, o que significa que seu trabalho é quebrar ou, pelo menos, contornar a proteção projetada e implementada pelo criptógrafo. Isto pode ser conseguido de duas maneiras, seja pela superação de uma hipótese fundamental de segurança ou através da exibição de um defeito específico no algoritmo ou protocolo considerado (Oppliger,

2005).

A definição do termo "*quebra*"[1] "*é executar criptanálise com êxito e, assim, conseguir descriptografar os dados ou executar alguma outra função de criptografia, sem inicialmente ter conhecimento do segredo que a função exige.*"

Este termo aplica-se a dados criptografados ou, mais geralmente, a um algoritmo criptográfico ou sistema de criptografia. Além disso, enquanto o uso mais comum é referir-se completamente quebrar um algoritmo, o termo também é usado quando for encontrado um método que reduz substancialmente o fator trabalho (Shirey, 2007).

A criptologia compreende estes dois campos, criptografia e criptanálise.[2]

1.1 Motivação

O Sistema Único de Saúde (SUS) brasileiro atribui aos Centros de Saúde, também denominados de unidades básicas de saúde, o papel de órgão provedor da atenção primária à saúde. Nos últimos anos, a estratégia preferencial de organização destes serviços tem sido o Programa de Saúde da Família.

O objetivo deste programa é o de melhorar a qualidade do serviço de saúde prestado à população por meio da aproximação entre equipes de saúde e a comunidade. A estratégia permite, sobretudo, uma mudança do paradigma de tratamento de doenças para o de promoção da saúde.

Ao mesmo tempo, a tecnologia tem se inserido cada vez mais em todas as áreas da Medicina, auxiliando em novas descobertas e em novas formas de oferecer uma assistência à saúde de melhor qualidade à população. A telemedicina é um ramo emergente da medicina que se aproveita de tecnologias de telecomunicações para tratar pacientes em locais remotos (Xiao e Chen, 2008).

Com o advento dos telefones celulares e outras tecnologias de telecomunicações móveis, a telemedicina está passando por um conjunto importante de mudanças.

Os dispositivos convencionais de telemedicina estão confiando cada vez mais em tecnologias de telecomunicações móveis de ponta, esse fenômeno é conhecido como telemedicina móvel. A popularidade de telemedicina móvel está crescendo rapidamente, principalmente devido à conveniência associadas com a tecnologia (Xiao e Chen, 2008).

A tecnologia da informação (TI) já está assistindo a um número sem precedentes de crimes de informática e a telemedicina celular não está imune a essas ameaças.

Apesar de sua importância para a organização e articulação dos sistemas de atenção primária, os programas de atenção domiciliar são normalmente conduzidos com pouco – ou nenhum – suporte de TI.

Com o objetivo de prover melhores ferramentas de TI para os profissionais de saúde, em 2004 foi iniciado o desenvolvimento de um sistema de captura de dados para dispositivos móveis, por exemplo "*smartphones*" e assistentes pessoais digitais (PDAs - "*Personal digital assistants*"), capaz de auxiliar o trabalho dos profissionais de saúde em campo.

Este trabalho faz parte do Projeto Borboleta, que inicialmente utilizou dispositivos móveis para auxiliar os profissionais da saúde nos atendimentos domiciliares, como uma ferramenta móvel de

[1] tradução do inglês *break*, no dicionário encontramos as seguintes definições: subst. fratura; ruptura; transgressão; verb. quebrar; romper; violar; interromper; cancelar; falir

[2] Alguns autores também acrescentam a Esteganografia ao conjunto de campos da criptologia (Wayner, 2008). Ela consiste em camuflar uma forma escrita em outra, a fim de mascarar o seu verdadeiro sentido.

apoio à coleta de informações para um Sistema Móvel de Prontuário Eletrônico, que integra novas tecnologias e está fortemente baseado no conceito de acompanhamento da situação clínica dos pacientes.

O conceito de sistema de Prontuário Eletrônico (PE) é mais amplo do que o de coleta de informação, pois implica armazenar os registros médicos (prontuários) dos pacientes em meios digitais (w. Bates *et al.*, 2003).

Este sistema, o sistema Borboleta, teve o seu escopo ampliado para atender outras áreas de um centro de saúde. Atualmente, está sendo validado pelos profissionais de saúde do Centro de Saúde Escola Butantã, parceiros no desenvolvimento do projeto e responsáveis por conduzir um Programa de Atenção Primária Domiciliar. Estes sistemas agregam valor ao serviço de saúde e, em larga escala, podem reduzir custos (Wang *et al.*, 2003).

A garantia de segurança e privacidade na telemedicina móvel é especialmente importante devido às seguintes razões:

- A natureza dos dados sendo trocados é fundamentalmente diferente. Senão tratada adequadamente, alguns dos dados poderiam até mesmo ameaçar a vida, porque dados errados podem facilmente levar a um diagnóstico fatal.

- Embora não seja fatal, o roubo ou a revelação de informações médicas podem ser usados para cometer crimes e discriminar uma pessoa para fins de vendas de seguros, locação, etc.

No Brasil, a Sociedade Brasileira de Informática em Saúde (SBIS) elaborou um conjunto de normas técnicas que devem ser seguidas para a construção de um PE [3].

No Projeto Borboleta a segurança da informação que trafega entre dispositivos móveis conectados sem fio se torna inerentemente necessário. Como os PDAs possuem baixo poder de processamento e conexões de pequena largura de banda, ainda exigem algoritmos de alta velocidade, alto nível de segurança e consumo de energia reduzida.

Por isso estudamos métodos baseados em curvas elípticas, em especial protocolos de acordo de chaves que possibilitam a combinação de chaves para utilização em algoritmos simétricos, garantindo a privacidade e segurança dos dados.

Escolhemos os esquemas de troca de chaves, pois após analisar os vários tipos de esquemas (protocolos de cifragem, assinatura e cifrassinatura) verificamos que este é um dos mecanismos com menor quantidade de chaves a ser armazenada.

As chaves de sessão geradas em cada acordo podem ser usadas como chave secreta de algoritmos simétricos para cifrar dados que trafegam por um canal inseguro.

O estudo de criptossistemas sem certificados e criptografia baseada em emparelhamentos compreende estudos desde áreas na matemática, como álgebra comutativa, geometria e topologia algébrica e funções analíticas, até definições de modelos de segurança que permite demonstrar a segurança dos protocolos.

1.2 Objetivos

O objetivo principal da dissertação é a construção de um mecanismo provedor de segurança e integridade das informações sigilosas dos dados dos prontuários médicos trafegados no sistema.

[3] Manual de Certificação para Sistemas de Registro Eletrônico em Saúde - disponível em http://www.sbis.org.br/certificacao/Manual_Certificacao_SBIS-CFM_2009_v3-3.pdf

De acordo com as normas , este tipo de mecanismo requer autenticação de ambos os lados além de ser eficiente, assim utilizamos um acordo de chaves baseado em criptografia sem certificados, demonstrado seguro para o problema computacional Diffie-Hellman bilinear (Lippold *et al.*, 2009).

1.3 Organização do Trabalho

Iniciando pelos conceitos iniciais de criptografia de chave pública, os esquemas de criptografia sem certificados, estudos relacionados com esquemas de acordo de chaves sem certificados, o modelo do oráculo aleatório, os principais problemas de complexidade que os criptossistemas se baseiam e os trabalhos correlatos no Capítulo 2.

No Capítulo 3 vamos apresentar os conceitos preliminares sobre as curvas elípticas, assim como a aritmética envolvida e como elas podem ser úteis a criptografia; no Capítulo 4, nós apresentamos emparelhamento bilinear, as aplicações de Weil, Tate e Ate, o algoritmo de Miller para calculá-los e aspectos computacionais.

No capítulo 5, apresentamos os protocolos de acordo de chaves autenticado estudados e a modificação proposta para implementação.

Em seguida, apresentamos os aspectos da implementação, como a biblioteca foi implementada, bem como as operações de corpos finitos e a implementação dos emparelhamentos bilineares.

No capítulo 7, descrevemos os experimentos realizados e os resultados obtidos dos experimentos realizados; no último capítulo finalizamos com a discussão das conclusões obtidas no trabalho e dos trabalhos futuros.

No Apêndice A, descrevemos os principais conceitos algébricos utilizados.

A idéia básica de divisores para compreender os emparelhamentos estão apresentados no Apêndice B.

Capítulo 2

Fundamentos de Criptografia de Chave Pública

Neste capítulo, vamos apresentar os fundamentos da criptografia assimétrica, ou de chave pública e os conceitos envolvidos na criação de um criptossistema. Começaremos com algumas definições preliminares.

Um algoritmo é uma máquina de Turing, um algoritmo eficiente e um adversário são algoritmos probabilísticos em tempo polinomial (Catalano *et al.*, 2005, pág.90),(Talbot e Welsh, 2006, sec.2.2 pág. 22).

Define-se as *funções negligenciáveis*[1] como aquelas funções que tendem a zero (assintoticamente), menor do que qualquer inversa de um polinômio. Isto é, uma função $\epsilon : \mathbb{N} \to \mathbb{R}$ é negligenciável, se para qualquer $c \in \mathbb{N}$ existe um $n_0 \in \mathbb{N}$ tal que temos $\epsilon(n) < 1/n^c$ para todo $n > n_0$ (Catalano *et al.*, 2005, cap.C pág.91).

Um algoritmo probabilístico, cujo tempo de execução é polinomial em $\log n$ é dito *eficiente* (Martin, 2008, pág.91).[2]

Um problema cujo algoritmo eficiente tem probabilidade negligenciável (baixa probabilidade) de obter sucesso, dada uma entrada aleatória, é dito *difícil* (Martin, 2008, def. 5.3 pág.91).

Então, um problema o qual existe um algoritmo eficiente que dada uma entrada aleatória tem alta probabilidade de obter sucesso é *fácil* (Martin, 2008, pág.92).

Para mostrar que um problema é "fácil" de resolver, basta dar um exemplo de um algoritmo eficiente prático para a sua solução.

Desse modo, um algoritmo de criptografia útil tem a propriedade de tanto para criptografar como descriptografar os dados é fácil com a chave certa, mas descriptografar os dados sem a chave correta é difícil (Talbot e Welsh, 2006, sec. 3.1 pág.53).

Os algoritmos assimétricos[3] contam com uma chave para criptografia e uma chave diferente, porém relacionada, para a decriptografia. A principal propriedade é que é computacionalmente inviável determinar a chave de decriptografia dado apenas o conhecimento do algoritmo de criptografia e da chave de criptografia.

[1] tradução do inglês "*negligible*"

[2] O uso de $\log n$, em vez de n é devido ao fato dos parâmetros e as chaves, que determinam as operações das funções criptográficas, serem tradicionalmente medidas em número de bits do parâmetro em vez do próprio tamanho do parâmetro.

[3] Este nome é utilizado também, porque a chave particular é distinta da chave pública

O artigo de Diffie, W. e Hellman, M. E. publicado em 1976 com o título "*New directions in cryptography*" introduziu o modelo de chave pública em que cada usuário possui um par de chaves (S, P) sendo S a chave particular[4] ou secreta, e P a chave pública. O par (S, P) é relacionado matematicamente de forma que:

1. Se M denota um texto legível, e $s()$ denota a aplicação da chave S, que transforma M em $s(M) = C$ então $p(C) = M$ onde $p()$ denota a aplicação da chave P. Ou seja S é a chave inversa da chave P – $p(s(M)) = M$;
2. O cálculo do par de chaves (S, P) é computacionalmente viável;
3. Porém, é computacionalmente difícil calcular S a partir do conhecimento de P;
4. Os cálculos de $p()$ e $s()$ são computacionalmente fáceis para quem conhece as chaves;
5. É computacionalmente difícil obter M a partir de $s(M)$.

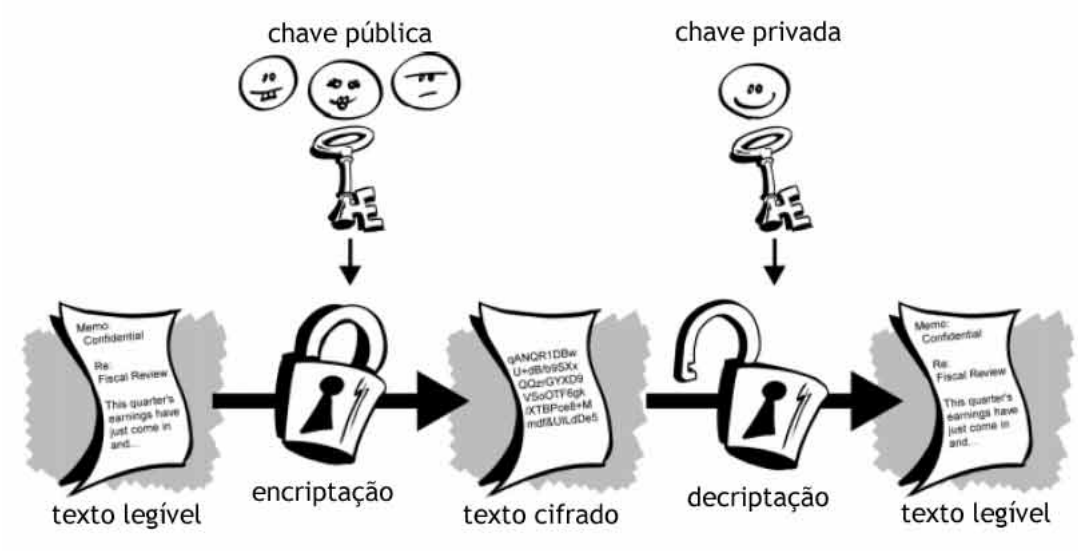


Figura 2.1: *Exemplo de criptografia assimétrica*

Com essa técnica, todos os participantes têm acesso às chaves públicas, que são mantidas em um registro público. As chaves particulares são geradas localmente por cada participante[5] e, portanto, nunca precisam ser distribuídas. Desde que a chave particular de um usuário permaneça protegida e secreta, a comunicação recebida está protegida.

O primeiro algoritmo de chave pública foi publicado no artigo inicial de Diffie e Hellman, é geralmente chamado de *acordo de chaves Diffie-Hellman*. A finalidade do algoritmo é permitir que dois usuários negociem uma chave com segurança, a qual pode então ser usada para a subsequente criptografia das mensagens. O próprio algoritmo é limitado ao acordo de valores secretos.

O algoritmo Diffie-Hellman depende, para sua segurança, da dificuldade de se calcular logaritmos discretos. Resumindo, pode-se definir o logaritmo discreto da seguinte forma:

[4] Esta chave é descrita na literatura como *chave privada*.
[5] tradução do inglês "*party*"

Primeiro, define-se uma raiz primitiva de um número primo p como aquele cujas potências módulo p geram todos os inteiros de 1 até $p-1$. Ou seja, se a é uma raiz primitiva do número primo p, então os números

$$a \bmod p, a^2 \bmod p, \cdots, a^{p-1} \bmod p$$

são distintos e consistem nos inteiros de 1 até $p-1$ em alguma permutação.

Para qualquer inteiro b e uma raiz primitiva a do número primo p, pode-se encontrar um expoente exclusivo j tal que

$$b \equiv a^j \pmod{p} \text{ onde } 0 \leq j \leq (p-1)$$

O expoente j é referenciado como o logaritmo discreto de b na base $a \bmod p$. Esse valor é expresso como $d\log_{a,p}(b)$.

Porém, no modelo de chave assimétrica existe o problema crítico de distribuição de chaves públicas de forma segura. A seguir apresentaremos alguns modelos alternativos de criptografia de chave pública que tentam minimizar este problema.

2.1 Modelos de criptografia de chave pública

O protocolo de acordo de chaves apresentado permite que duas pessoas combinem uma chave secreta. O acordo de chave se dá por meio de um canal de comunicação, porém sem garantia de autenticação nem de origem, nem de destino. Essa dupla ausência de autenticidade possibilita uma ofensiva ao sistema, chamado ataque do "homem no meio", no qual um impostor, no meio da linha de comunicação, intercepta as mensagens alterando-as. Esse falso usuário personifica o receptor perante o emissor e ainda se faz passar pelo emissor perante o receptor. Dessa forma, o impostor se torna capaz de decifrar as mensagens trocadas.

Sob o protocolo de Diffie e Hellman (1976), o ataque do "homem no meio" torna-se viável porque as chaves públicas não estão vinculadas às respectivas identidades. Uma pequena modificação nesse protocolo produz um modelo muito útil, aparentemente resistente, que é utilizado até os dias atuais. A modificação requer a inclusão de uma chave secreta adicional, segredo que só um terceiro conhece uma entidade de confiança entre os interlocutores.

A partir de então, o acordo de chaves de Diffie-Hellman pode ser derivado para algoritmos de criptografia de chave pública. Foi sugerido por Kohnfelder (1978) a criação de um mecanismo de certificação digital com o objetivo de associar univocamente a chave pública a uma entidade que utiliza uma estrutura hierárquica de autoridades certificadoras capaz de garantir adequadamente a posse de uma dada chave pública.

Este mecanismo funciona muito bem em organizações abertas como a internet. Porém, neste modelo, existe pouca flexibilidade para troca de chaves secretas, pois a chave pública emitida pela infra-estrutura que gerencia chaves públicas depende do valor secreto, o certificado só pode ser gerado e distribuído depois do usuário gerar (ou recriar) o par de chaves; a seguir, a autoridade de confiança necessita emitir, assinar e distribuir o novo certificado.

Portanto a infra-estrutura de chaves públicas (ICP/PKI) requer gerenciamento de certificados que podem se tornar inválidos dentro do período de validade, devido ao comprometimento do valor

secreto possibilitando a falsificação do usuário.

2.2 Modelo criptografia baseado em identidade

Em 1984 um modelo de criptografia de chave pública baseada em identidades foi proposto por Shamir (1985), em que identificadores como nome, endereço de e-mail ou CPF, funcionam como chave pública. Desse modo, automaticamente se cria um vínculo entre chave pública e respectivo dono, o que dispensa a necessidade de certificados e infra-estruturas de chave pública.

O primeiro esquema de cifragem e decifragem que pôde ser demonstrado seguro e ao mesmo tempo eficiente, do ponto de vista computacional, foi o esquema de Boneh e Franklin (2001), chamado IBE, de "*Identity Based Encryption*". Para aplicações em telefonia móvel, por exemplo, as mensagens poderiam ser transmitidas com sigilo de forma transparente aos usuários: o remetente precisaria apenas conhecer o identificador do destinatário: o número telefônico, por exemplo.

Um problema que existe nesta idéia é o conhecimento da chave secreta pela autoridade central, ou Gerador de chaves particulares (PKG), sendo necessária uma confiança total pelo usuário, o que exige uma série de cuidados do ponto de vista prático e legal. Por outro lado, não é necessária toda a infra-estrutura hierárquica de autoridades para o gerenciamento das chaves, tornando o modelo mais simples e adequado para organizações onde a hierarquia é natural e seus limites são bem controlados.

2.3 Criptografia sem certificado

A criptografia sem certificados combina o melhor da criptografia tradicional de chave pública e o paradigma baseado em identidade, porque não há necessidade de obter e verificar os certificados, e as chaves particulares não estão sob custódia[6] do centro de geração de chaves (KGC - "*Key Generator Center*") (Al-Riyami e Paterson, 2003).

A criptografia baseada em identidade (ID-based) evita os custosos certificados, mas torna a geração da chave particular, uma operação privilegiada que só uma autoridade confiável pode executar. Essa custódia de chaves é desejável em algumas aplicações, mas em muitos contextos, é totalmente inaceitável (Boneh e Franklin, 2001).

Apesar da custódia parecer inevitável na criptografia baseada em identidade, não se descarta a possibilidade de haver um criptossistema de chave pública livre de custódia e dos problemas de infra-estrutura de chave pública (ICP).

Em um sistema de criptografia de chave pública, apenas a autenticação é capaz de completar a tarefa da criptografia. Por exemplo, somente o destinatário de uma mensagem cifrada pode decifrar e ler a mensagem em texto legível.

O problema da custódia, no paradigma baseado em identidade, é causado pelo fato de que o gerador de chave particular (KGC) sempre pode gerar a chave particular correspondente a uma sequência especifica de uma identidade, que é suficiente para fazer a decriptografia.

Por outro lado, a autenticação é feita antes que se possa obter tal chave particular. Enquanto a chave particular é necessária na decodificação, a autenticação é executada implicitamente e não há necessidade de recorrer a meios especiais como um certificado para provar que ela ocorreu.

[6] (*lat custodia*) sf *Dir* guarda ou detenção de coisa alheia, que se administra e conserva, até a entrega ao seu dono legítimo (Gregorim *et al.*, 2002).

Tais observações levam a uma possível abordagem para contornar o problema da custódia sem o uso do certificado. Introduz-se um componente extra na tarefa criptográfica para exigir não só a chave particular do sistema criptográfico baseado em identidade, mas também algo fora do conhecimento do centro gerador de chave (KGC). Esta é a idéia da criptografia (de chave pública) sem certificados.

Uma pesquisa foi realizada por Dent (2008), com mais de vinte esquemas de criptografia sem certificado que incide sobre os diferentes modelos de segurança e a eficiência dos respectivos esquemas.

Nos esquemas de criptografia sem certificados, há três segredos por participante:

- A chave emitida pelo centro de geração de chaves é chamada de "chave particular parcial". Nós assumimos que esta chave é baseada na identidade, embora não necessariamente têm que ser *ID-based*.
- A chave particular x_{ID} gerada pelo usuário, é chamada de "valor secreto".
- O valor efêmero escolhido aleatoriamente para cada sessão.

2.4 Protocolos de acordo de chaves

Em criptografia, um protocolo de acordo de chave é um protocolo através do qual dois ou mais participantes podem combinar uma chave secreta, usando canais públicos.

Se feito corretamente, exclui que terceiros indesejados forcem uma escolha de chave no acordo dos participantes. Por isso, fornece um meio seguro e eficiente para duas partes se comuniquem por um canal vulnerável e controlado por um adversário.

Os protocolos de acordo de chaves com autenticação fazem o mesmo, porém com garantia de autenticidade dos participantes.

Apresentaremos o modelo do oráculo aleatório e alguns problemas utilizados para realizar provas de segurança dos protocolos criptográficos.

2.5 Modelo do oráculo aleatório e suas implicações

O nosso modelo básico de comunicação supõe duas entidades, Alice e Beto, trocando mensagens transmitidas num canal inseguro; isto é, um canal passível de leitura e escrita por um intruso, Carlos.

Os métodos de Carlos podem ser a simples escuta, um "grampo", que chamamos de ataque passivo, ou até a modificação, repetição e injeção de mensagens com objetivos variados como, por exemplo, passar-se por Alice ou Beto para obter acesso a serviços não autorizados; esses são os chamados ataques ativos.

O modelo do oráculo aleatório foi o primeiro modelo de segurança a ser introduzido na comunidade da criptografia (Bellare e Rogaway, 1993; Fiat e Shamir, 1987).

Os oráculos aleatórios são uma abstração matemática utilizada em provas de criptografia, que são normalmente utilizados quando nenhuma função conhecida implementável fornece as propriedades matemáticas exigidas pela prova.

Formalmente, um oráculo aleatório é um mapeamento $R : \{0,1\}^* \rightarrow \{0,1\}^\infty$, onde cada bit de $R(x)$ é escolhido independentemente e de modo uniforme, para todo x. A notação $\{0,1\}^\infty$ refere-se a cadeias suficientemente longas, geradas pelo oráculo (Goya, 2006, sec. 3.8 pág.40).

Uma prova de segurança no modelo de oráculo aleatório irá fornecer um forte argumento a favor da segurança do esquema.

A prova geral mostra que um sistema ou um protocolo é seguro, demonstrando que um invasor deve exigir comportamento impossível do oráculo, ou resolver algum problema matemático considerado difícil, a fim de quebrar o protocolo (Bellare e Rogaway, 1993, sec.5.2 pág.12).

A prova que não utiliza tais oráculos aleatórios é dita usar o modelo padrão (Stern *et al.*, 2002).

Além disso, as provas no modelo do oráculo aleatório sob uma hipótese computacional fraca podem ser de interesse mais prático que as provas no modelo padrão com uma hipótese computacional forte.

Como dito acima, a eficiência raramente se encontra com a prova de segurança. Mais precisamente, nenhum dos esquemas mais eficientes em suas categorias foi provado seguro no modelo padrão (Catalano *et al.*, 2005, sec.2.3 pág.137).

No entanto, alguns deles admitem validações de segurança com base em hipóteses ideais: o modelo do oráculo aleatório é o mais amplamente aceito.

Acredita-se que os verdadeiros oráculos aleatórios são impossíveis de implementar, então quando uma prova é obtida neste modelo, os oráculos aleatórios são substituídos por funções cujo comportamento é similar o suficiente para que a segurança do sistema ainda pareça plausível (Chen *et al.*, 2007, pág.3).

Na prática, os oráculos aleatórios são normalmente utilizados para modelar as funções de hash criptográficas em sistemas em que fortes hipóteses de aleatoriedade são necessárias na saída da função hash.

Neste modelo, o chamado modelo do oráculo aleatório, a função de hash pode ser formalizada através de um oráculo que produz um valor verdadeiramente aleatório para cada nova consulta. Claro que, se a mesma consulta é feita duas vezes, respostas idênticas serão obtidas. Este é precisamente o contexto da teoria da complexidade relativizada com "oráculos".

Muitos esquemas de criptografia usam uma função de hash H (tal como MD5 quebrado por (Wang e Yu, 2005), ou os padrões americanos SHA-1 (institute of standards e technology, 2002), SHA-256, SHA-384 e SHA-512 (institute of standards e technology, 2001)). Este uso de funções de hash foi originalmente motivado pelo desejo de assinar as mensagens longas com uma assinatura única e curta.

Com a finalidade de alcançar a irretratabilidade[7], um requisito mínimo na função de hash é a impossibilidade do assinante encontrar duas mensagens diferentes que forneçam o mesmo valor de hash. Esta propriedade é chamada resistência à colisão.

Mais tarde, concluiu-se que as funções de hash são um ingrediente essencial para a segurança dos esquemas de assinatura e assim da maioria dos esquemas de criptografia (Catalano *et al.*, 2005).

Para obter argumentos de segurança, mantendo a eficiência dos projetos que usam funções de hash, alguns autores sugeriram a utilização da hipótese de que H se comportasse como uma função aleatória.

Sobre esse modelo, ninguém jamais foi capaz de fornecer uma contradição convincente para a sua

[7] *sf* que não se pode retratar; irrevogável (*lat irrevocabile*) *adj m+f* não revogável, que não se pode anular.

validade prática, mas apenas contra-exemplos teóricos nos projetos evidentemente incorretos para a finalidade prática (Canetti *et al.*, 2004, pág. 13), ou noções de segurança artificiais (Bellare *et al.*, 2003; Nielsen, 2002).

Portanto, este modelo tem sido fortemente aceito pela comunidade, e é considerado como um modelo bom, em que a analise de segurança fornece uma boa prova do nível atual de segurança.

Mesmo que ele não forneça uma prova formal de segurança (como no modelo padrão, sem qualquer hipótese ideal), argumenta-se que as provas neste modelo garantam a segurança da concepção global do esquema, desde que a função de hash não tenha nenhuma fraqueza, daí o nome de "argumentos de segurança".

Este modelo também pode ser visto como uma restrição às capacidades do adversário. Na verdade, isso simplesmente significa que o ataque é genérico sem considerar qualquer "instanciação" particular das funções de hash.

Portanto, um ataque real deveria necessariamente usar uma fraqueza ou uma característica específica da função de hash. Facilmente solucionado pela substituição da função de hash por outra que elimina este ataque.

Por outro lado, supondo que a resistência à adulteração de alguns dispositivos, tais como *smart-cards*, o modelo do oráculo aleatório é equivalente ao modelo padrão que, apenas, exige a existência de funções pseudo-aleatórias (Goldreich *et al.*, 1986; M'Raïhi *et al.*, 1999).

Como consequência, quase todas as atuais organizações de padronização exigem projetos comprovadamente seguros, pelo menos nesse modelo, graças à validação de segurança de protocolos práticos muito eficientes.

A seguir, apresentam-se os principais problemas utilizados para demonstração de segurança no modelo do oráculo aleatório.

2.6 Problemas de interesse

Seja um grupo cíclico $\mathbb{G}$ de ordem prima q (como o grupo finito $(\mathbb{Z}_q, +)$, um subgrupo de $(\mathbb{Z}_p, \times)$ para $q|p-1$, de uma curva elíptica[8]), e um gerador $\mathbf{g}$ (i.e. $\mathbb{G} = \langle \mathbf{g} \rangle$).

Nota-se em negrito (como $\mathbf{g}$) qualquer outro elemento do grupo $\mathbb{G}$, para distingui-lo de um escalar $x \in \mathbb{Z}_q$. Mas tal $\mathbf{g}$ poderia ser um elemento em $\mathbb{Z}_p^*$ ou um ponto de uma curva elíptica, de acordo com a configuração.

Acima, citou-se um grupo de $\mathbb{G}$ "adequado". Nesse grupo, alguns dos seguintes problemas necessitam ser difíceis para resolver (usando a notação aditiva):

- Problema do Logaritmo Discreto (DLP): Dado $y \in \mathbb{G}$ e $\mathbf{g}$ um gerador de $\mathbb{G}$, encontrar um valor $x \in \mathbb{Z}_q$, tal que $y = \mathbf{g}^x$.
- Problema Diffie-Hellman Computacional (CDHP): Dados dois elementos $x\mathbf{g}, y\mathbf{g} \in \mathbb{G}$, onde $x, y \in \mathbb{Z}_q^*$ e $\mathbf{g}$ um gerador do grupo $\mathbb{G}$: calcular o valor $z = xy.\mathbf{g}$, definido como $z = DH(x, y)$.
- Problema de Decisão de Diffie-Hellman (DDHP): Dados $x\mathbf{g}, y\mathbf{g}, z\mathbf{g} \in \mathbb{G}$, decidir se $z = DH(x, y)$ (ou equivalentemente se $z = xy \bmod q$).

[8] Veremos as idéias de curva elíptica no próximo capítulo

Os problemas estão ordenados do problema mais forte para o mais fraco. Além disso, pode-se observar que todos são "auto-redutíveis aleatórios", o que significa que qualquer instância pode ser reduzida a uma instância distribuída uniformemente: por exemplo, dado um elemento y o qual deseja-se calcular o logaritmo discreto de x na base g, pode-se escolher um $z \in \mathbb{Z}_q$ aleatório, e calcular $z = zy$.

O elemento z é, portanto, distribuído uniformemente no grupo, e o logaritmo discreto $\alpha = \log_g = z$ leva a $x = \alpha/z \bmod q$. Como consequência, há apenas casos de complexidade média. Assim, a capacidade de resolver um problema por uma fração considerável de casos em tempo polinomial é equivalente a resolver qualquer instância em tempo esperado polinomial.

O PLD pode ser resolvido, evidentemente, por busca exaustiva (por força bruta).

Para tal, calculam-se sistematicamente todos os pares $(x; g^x)$ até encontrar um par onde $y = g^x$. Esse algoritmo funciona em espaço $O(1)$, mas infelizmente em tempo $O(n)$, ou seja, $O(2^k)$ (exponencial).

Outras técnicas fazem uso da estrutura do grupo onde se quer calcular o logaritmo discreto. Um dos algoritmos dessa classe é o chamado *cálculo de índices*, aplicável na solução do PLD sobre o grupo multiplicativo de um corpo finito.

A complexidade do algoritmo básico do cálculo de índices é *subexponencial*, $O(2^{\sqrt{k \log k}})$, sendo muito mais eficiente do que os algoritmos genéricos exponenciais (conquanto ainda intratável para valores suficientemente grandes de n).

O algoritmo mais rápido conhecido para resolver o PLD é uma variante do cálculo de índices chamada *crivo de corpo numérico generalizado* (em inglês *generalized number field sieve*), e possui complexidade assintótica de operação $O(2^{\sqrt[3]{k \log^2 k}})$ (Menezes *et al.*, 1996, sec. 3.6).

Uma variante do problema Diffie-Hellman definida recentemente por Okamoto e Pointcheval (2001), o chamado *Problema Gap Diffie-Hellman* (GDHP), onde deseja-se resolver o problema CDHP por meio de um acesso a um oráculo DDHP.

Nesse contexto, pode-se notar as seguintes propriedades sobre os problemas acima:

$$DLP \longrightarrow CDHP \longrightarrow DDHP, GDHP,$$

onde $X \rightarrow Y$ significa que o problema X é pelo menos tão difícil quanto o problema Y. No entanto, até hoje, ninguém sabe como resolver nenhum deles sem quebrar o próprio problema do logaritmo discreto (Cash *et al.*, 2008, sec.2 pág.6).

Em alguns casos, é perceptível a correspondência entre a existência da qualificação de um dos problemas computacionais e a destreza de um adversário para atacar um sistema de chave pública.

O CDHP, por exemplo, é modelado conforme as informações que um adversário observa numa troca de chaves Diffie-Hellman e de quais informações o adversário deseja obter no intuito de atacar o sistema.

No entanto, em outros casos a correspondência não é tão clara.

O fato da força dos protocolos sem certificados, que serão apresentados no Capítulo 5, é pelo menos tão forte quanto certos problemas computacionais pode não ser claro, devido à complexidade dos algoritmos.

Por exemplo, o protocolo é bom, pois sabe-se que existem provas que quebrá-lo é pelo menos tão difícil quanto resolver problemas computacionais que acredita-se serem difíceis.

Para provar que um algoritmo criptográfico é pelo menos tão forte quanto um determinado

problema computacional; a técnica típica é assumir que um adversário que possua um algoritmo capaz de atacar o algoritmo criptográfico de interesse. E então demonstra-se que ele pode usar esse algoritmo de ataque para construir um algoritmo que resolve o problema computacional de interesse.

Assim, se acreditamos que a resolução do problema computacional é difícil, também será difícil atacar o algoritmo de criptografia.

Note que isso não mostra que o algoritmo de criptografia é realmente seguro, se existe a possibilidade de resolver o problema computacional relacionado, então os algoritmos criptográficos podem ser atacados.

Então, para demonstrar que a troca de chaves Diffie-Hellman é pelo menos tão forte quanto o CDHP, mostra-se que um atacante capaz de atacar a troca de chaves Diffie-Hellman pode usar o algoritmo dele para efetuar o ataque e resolver o CDHP.

Isso não seria mostrar que a troca de chaves Diffie-Hellman é segura, mas mostra que se um atacante pode atacar a troca de chaves Diffie-Hellman, então ele também poderia realizar algo que se acredita ser difícil (Katz e Lindell, 2007, sec.3.1.3 pág.58).

Se acreditarmos que o CDHP é realmente difícil de resolver, então tal prova também nos convence que o ataque à troca de chaves Diffie-Hellman também seja difícil.

A área de *segurança provável*, ou *segurança demonstrável*, compreende uma subárea de pesquisa de protocolos criptográficos. Este trabalho utiliza-se destes conhecimentos apenas como critério de avaliação dos protocolos a serem utilizados.

2.7 Trabalhos correlatos

Uma visão de quase vinte protocolos de acordo de chaves baseados em identidade foi compilado por Cheng *et al.* (2007), eles fornecem provas de segurança para dois dos protocolos.

Muitos sistemas baseados em identidade garantem total privacidade para ambas as partes, desde que o KGC não conheça nenhum dos segredos efêmeros usados no cálculo da chave de sessão.

Mas, como Krawczyk (2005) aponta, o vazamento de chaves efêmeras não deve ser negligenciado, porque elas são geralmente pré-computadas e não são armazenadas em memória segura.

No contexto de protocolos de acordo de chave baseado em identidade, isto significa que, assim como os vazamentos de partes da chave efêmera, um KGC mal-intencionado é capaz de calcular a chave de sessão.

Uma visão geral dos esquemas atuais de acordo de chaves sem certificados foi compilada por Swanson (2008).

Os esquemas de acordo de chave sem certificado tentam proporcionar privacidade, ainda que parte do segredo efêmero seja enviada para o centro gerador de chaves, ou mesmo que o centro de geração de chaves interfira ativamente nas mensagens que são trocadas. Por exemplo, realizando um "*ataque do homem no meio*"[9].

O primeiro esquema de acordo de chaves sem certificado foi publicado por Al-Riyami e Paterson (2003) como uma nota do esquema de encriptação sem certificados.

[9] ataque onde um interlocutor tenta ler todas as mensagens trocadas no meio inseguro as interceptando e modificando

No entanto, eles não forneceram um modelo de segurança para os principais esquemas de acordo sem certificado, nem uma prova de segurança para o protocolo.

Outros esquemas de acordo de chave sem certificados foram publicadas por Mandt e Tan (2006) e aperfeiçoado por Xia *et al.* (2008), Wang Shengbao (2006), e Shao (2005), mas os respectivos autores deram apenas argumentos heurísticos para justificar a segurança dos seus protocolos.

Swanson (2008) também analisou estes esquemas sem certificados e mostrou ataques genéricos que quebram as noções de segurança reivindicada pelos respectivos autores.

Em 2009, Lippold *et al.* (2009) apresentou o primeiro protocolo de acordo de chaves sem certificado com demonstrado seguro no modelo proposto por Cash *et al.* (2008).

No Capítulo 5 será apresentado mais detalhes sobre os principais protocolos.

Capítulo 3

Conceitos preliminares

Neste capítulo, iremos apresentar uma introdução sobre a aritmética de curvas elípticas e demais conceitos preliminares para o Capítulo 4 e para a criptografia de curvas elípticas.

Para os leitores pouco familiarizados com a Álgebra, apresentamos as idéias necessárias para melhor compreensão deste capítulo no Apêndice A.

3.1 Curvas Elípticas

Em matemática, as curvas elípticas são definidas mediante equações cúbicas. Foram utilizadas para provar o último teorema de Fermat e se empregam também em criptografia e na fatoração de inteiros.

As curvas elípticas são "*regulares*", ou pode-se dizer "*não-singular*", o que significa que não têm "*cúspides*" nem auto-intersecções, então pode-se definir uma operação binária para o conjunto de seus pontos de uma maneira geométrica natural, o que faz deste conjunto um grupo abeliano (Hancock, 2000, pág. 111).

As curvas elípticas sobre o corpo dos números reais vêm a ser dadas por equações na forma de *Weierstraß*[1] como: $y^2 = x^3 - x$ e por $y^2 = x^3 - x + 1$.

As curvas elípticas podem ser definidas sobre qualquer corpo $\mathbb{K}$; a definição formal de uma curva elíptica é a de uma curva algébrica projetiva não singular sobre $\mathbb{K}$ de gênero 1[2].

Se a característica de $\mathbb{K}$ é diferente de 2 e 3 ($\text{car}(\mathbb{K}) \neq 2, 3$), então toda curva elíptica sobre $\mathbb{K}$ pode ser escrita na forma:

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

onde a e b são elementos de $\mathbb{K}$ tais que o polinômio do lado direito $x^3 - ax - b$ não tenha nenhuma raiz dupla. Se a característica é 2 ou 3 existirão mais termos como apresentado em Jeffrey Hoffstein (2008) (Sec.5.7 pág.308) (Ian F. Blake, 1999).

3.2 Aritmética de curvas elípticas

Definição 3.1. *(Silverman, 1986, sec.III.1 pág.46)* Uma *curva elíptica* E sobre um corpo $\mathbb{K}$ é

[1] Em homenagem ao matemático alemão Karl Weierstraß(1815-1897)

[2] as curvas de gênero > 1, são denominadas de *curvas hiperelípticas*, e seu estudo está fora do escopo deste trabalho

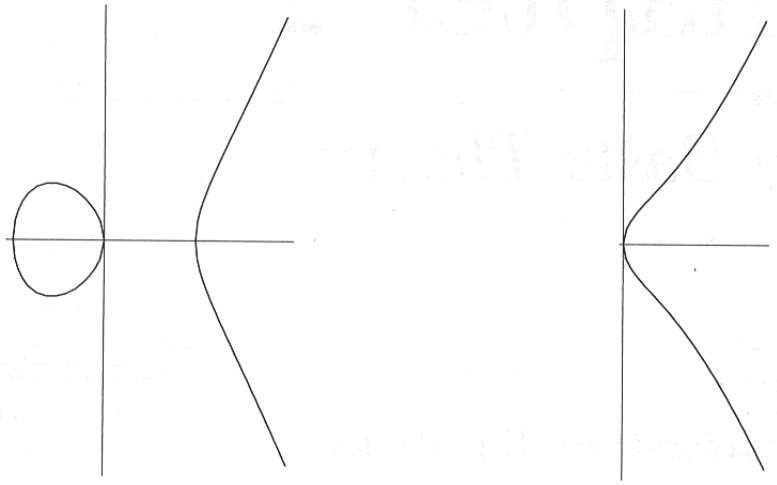


Figura 3.1: *Exemplos de curvas elípticas:(a) curva não-singular (b) curva singular*

definida pela equação

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \qquad \textit{(Equação de Weierstraß)}$$

onde $a_1, a_2, a_3, a_4, a_6 \in \mathbb{K}$ e $\Delta \neq 0$, onde Δ é o discriminante de E e é definido como se segue:

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$$

$$\begin{aligned} d_2 &= a_1^2 + 4a_2 \\ d_4 &= 2a_4 + a_1a_3 \\ d_6 &= a_3^2 + 4a_6 \\ d_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 + a_4^3 \end{aligned}$$

Por isso, uma curva elíptica com discriminante $\Delta = 0$ é chamada de *singular*, e de *não-singular* caso contrário.

Para a curva $y^2 = x^3 - ax - b$ considerada na Seção 3.1 ($\text{car}(\mathbb{K}) \neq 2, 3$), o discriminante Δ será $4a^3 + 27b^2$. Uma forma alternativa à fórmula de Cardano e Tartaglia se segue:

Demonstração. Para verificar o resultado, suponha $f(x) = x^3 - s_1x^2 + s_2x - s_3$, *e sabendo que:*

$$\begin{aligned} s_1 &= \alpha_1 + \alpha_2 + \alpha_3 \\ s_2 &= \alpha_1\alpha_2 + \alpha_1\alpha_3 + \alpha_2\alpha_3 \\ s_3 &= \alpha_1\alpha_2\alpha_3 \end{aligned}$$

e, também:

$$\begin{aligned} p_1 &= s_1 \\ p_2 &= \alpha_1^2+\alpha_2^2+\alpha_3^2 = (\alpha_1+\alpha_2+\alpha_3)^2 - 2(\alpha_1\alpha_2+\alpha_1\alpha_3+\alpha_2\alpha_3) \\ &= s_1^2 - 2s_2 \\ p_3 &= \alpha_1^3+\alpha_2^3+\alpha_3^3 = \textstyle\sum_{j=1}^{3}(s_1\alpha_j^2 - s_2\alpha_j + s_3) \\ &= s_1p_2 - s_2p_1 + 3s_3 \\ p_4 &= s_1p_3 - s_2p_2 + s_3p_1 \\ &= s_1^4 - 4s_1^2s_2 + 4s_1s_3 + 2s_2^2 \end{aligned}$$

pois,

$$\delta = \det\begin{vmatrix} 1 & 1 & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 \\ \alpha_1^2 & \alpha_2^2 & \alpha_3^2 \end{vmatrix}\begin{vmatrix} 1 & \alpha_1 & \alpha_1^2 \\ 1 & \alpha_2 & \alpha_2^2 \\ 1 & \alpha_3 & \alpha_3^2 \end{vmatrix} = \det\begin{vmatrix} 3 & p_1 & p_2 \\ p_1 & p_2 & p_3 \\ p_2 & p_3 & p_4 \end{vmatrix} = \delta = -4s_1^3s_3+s_1^2s_2^2+18s_1s_2s_3-4s_2^3-27s_3^2.$$

então

$$\delta = -4s_1^3s_3 + s_1^2s_2^2 + 18s_1s_2s_3 - 4s_2^3 - 27s_3^2$$

Na fórmula acima, se $s_1 = 0$, o discriminante fica

$$\delta = -4s_2^3 - 27s_3^2.$$

Se $f(x) = x^3 + ax^2 + bx + c \in \mathbb{K}[x]$ e $\mathbb{K}$ tem característica $\neq 2, 3$, então a mudança de variável $x = y - a/3$ (que não muda o grupo de Galois) transforma $f(x)$ em $g(x) = y^3 + a_2y + a_3$, com $a_2 = (b - a^2/3)$ e $a_3 = (c - a^3/27) - ab/3 + a^3/9$.

□

Se $\mathcal{L}$ é um corpo estendido de $\mathbb{K}$, então o conjunto de pontos $\mathcal{L}$ sobre E é

$$E(\mathcal{L}) : \{(x, y) \in \mathcal{L} \times \mathcal{L} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \mathcal{O}$$

onde $\mathcal{O}$ é o ponto no infinito (Silverman, 1986, sec.III.2 pág.57).

Sejam $E/\mathbb{K}$ e $E'/\mathbb{K}$ duas curvas elípticas. Se E e E' forem isomorfas sobre $\mathbb{K}$, então elas possuem o mesmo invariante j. Por outro lado, se $j(E) = j(E')$, então E e E' são isomorfos sobre o fecho algébrico de $\mathbb{K}$.

Definição 3.2. *(Martin, 2008, def. 3.13)* Se $E/\mathbb{F}$ é uma curva elíptica na forma normal de Weierstraß $y^2 = x^3 + ax + b$, diz-se que uma curva elíptica $E'/\mathbb{F}$ na forma normal de Weierstraß $y^2 = x^3 + a'x + b'$ é *isomorfa* sobre $\mathbb{F}$ se existe $u \in \mathbb{F}^*$ com $a' = u^4a$ e $b' = u^6b$.

Esta definição é motivada pelo isomorfismo do reticulado subjacente ao plano complexo definido pelo número inteiro múltiplo de dois períodos da função $\wp$ de Weierstraß $\{\omega_1, \omega_2\}$ (Martin, 2008, cap.3 pág.60).

Esse reticulado com períodos de $\{\omega_1, \omega_2\}$ é isomorfo a outro reticulado , se os dois períodos diferem-se pela mesma constante, ou o segundo reticulado é definido por múltiplos inteiros dos períodos $\{c \cdot \omega_1, c \cdot \omega_2\}$ para algum $c \in \mathbb{R}$.

As curvas elípticas isomorfas vêm da função $\wp$ definida sobre tais reticulados isomórficos.

O discriminante, tal como definido na Seção 3.2, só diz quando a parte cúbica de uma curva elíptica não tem raízes repetidas, e pode haver muitas curvas elípticas não isomórficas com o mesmo discriminante. Uma quantidade diferente é necessária para distinguir entre as curvas isomórficas.

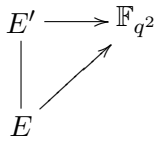
3.3 Torções em curvas elípticas

A observação que o j-invariante não altera-se sob a mudança de variáveis $a \to v^2a$ e $b \to v^3b$ leva à seguinte definição.

Definição 3.3. Seja $E/\mathbb{F}_q : y^2 = x^3+ax+b$ uma curva elíptica e $v \in \mathbb{F}_q^*$ um não resíduo quadrático em $\mathbb{F}_q^*$.

Então, $E'/\mathbb{F}_q : y^2 = x^3 + v^2ax + v^3b$, é chamado de *torção quadrática* de E.

Neste caso, E é isomorfo a E' sobre uma extensão de grau 2 de $\mathbb{F}_q$, mas não sobre $\mathbb{F}_q$ em si.



Em particular, um isomorfismo entre as curvas elípticas é uma isogenia de grau 1, que é uma isogenia inversível (Silverman, 1986, sec.III.4 pág.70).

3.3.1 Torções de alto grau

Para algumas curvas $E/\mathbb{F}_q$ é possível criar torções que não sejam a torção quadrática.

Nestes casos, tem-se $E' : y^2 = x^3 + a'x + b'$ onde $a' = v^{4/d}a$ e $b' = v^{6/d}b$, e v é uma raiz de grau d, mas não uma raiz menor que o grau d sobre $\mathbb{F}$ (assim, por exemplo, uma quarta raiz é uma raiz quarta, mas não uma raiz quadrada), o que pode-se chamar de uma *torção de grau d*.

Tais torções são isomorfas a E sobre $\mathbb{F}_{q^d}$, uma extensão de grau d para $\mathbb{F}_q$.

As torções possíveis, tanto quadrática e de grau maior, estão resumidas nas Tabelas 3.1 e 3.2.

Grau de Torção d	Forma de E	Forma de E'
2	$y^2 = x^3 + ax + b$	$y^2 = x^3 + v^2ax + v^3b$
3	$y^2 = x^3 + b$	$y^2 = x^3 + vb$
4	$y^2 = x^3 + ax$	$y^2 = x^3 + vax$
6	$y^2 = x^3 + b$	$y^2 = x^3 + vb$

Tabela 3.1: *Curvas elípticas e suas torções*

Grau de Torção d	Pontos típicos de E	Pontos correspondentes a E'
2	(x, y)	$(vx, v^{3/2}y)$
3	(x, y)	$(v^{1/3}x, v^{1/2}y)$
4	(x, y)	$(v^{1/2}x, v^{3/4}y)$
6	(x, y)	$(v^{1/3}x, v^{1/2}y)$

Tabela 3.2: *Pontos nas torções de curvas elípticas*

Em cada um destes casos, também deve haver $q \equiv 1 \pmod{d}$ para uma torção existir.

O Capítulo 4 apresentará as aplicações $\varphi_d : E' \to E$, onde d é o grau de uma torção, usadas na criação de estruturas úteis para a implementação de algoritmos baseados em emparelhamento.

Note que essas aplicações aumentam a dimensão de sua saída por um fator de d, de modo que se as entradas são elementos de algum $\mathbb{F}_d$, então as saídas são elementos de algum $\mathbb{F}_{q^d}$.

Definição 3.4. *(Martin, 2008, def.3.16 pág.62)* Seja $E/\mathbb{F}_q$ uma curva elíptica, n um inteiro relativamente primo a q e P um ponto de ordem n em $E(\mathbb{F}_q)$.

Uma *aplicação de distorção*[3] com relação a (ou para) P é um endomorfismo ϕ que mapeia o ponto P para um ponto $\phi(P)$ que é linearmente independente a P.

Outro ponto de vista útil é que essa aplicação de distorção é um endomorfismo não-racional[4].

As aplicações de distorção criam estruturas utilizadas para a implementação de muitos algoritmos sem certificados. Isto será discutido no próximo capítulo.

No entanto, sua aplicação é essencialmente limitada para curvas supersingulares, definidas na Seção 3.6.2, como as duas propriedades descritas a seguir.

Propriedade 3.1. *(Verheul, 2004, teo.6 pág.199)* Seja $E/\mathbb{F}_q$ uma curva elíptica supersingular com $P \in E(\mathbb{F}_q)[n]$. Se n é relativamente primo à característica de $\mathbb{F}_q$, então sempre existe uma aplicação de distorção com relação a P.

Propriedade 3.2. *(Verheul, 2004, teo.11 pág.206)* Seja $E/\mathbb{F}_q$ uma curva elíptica ordinária e seja $P \in E(\mathbb{F}_q)[n]$. Se n é relativamente primo à característica de $\mathbb{F}_q$ e $E[n] \in E(\mathbb{F}_q)$, então não pode existir uma aplicação de distorção com relação a P.

3.4 Lei de grupos de curvas elípticas

Considere uma curva elíptica:

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

O inverso do ponto $P = (x, y)$ é o ponto espelhado no eixo x, por exemplo $-P = (x, -y)$.

Geometricamente, a adição é descrita como: Dado $P, Q \in E(\mathbb{F}) = E$, $Q \neq \pm P$ e $P, Q \neq \mathcal{O}$. Desenha-se a linha l_1 secante através dos pontos P e Q. Esta linha irá interseccionar a curva E exatamente no ponto $R' = P \times Q$.

Desenhe a linha l_2 paralela ao eixo y passando por R'. A linha l_2 irá interseccionar a curva exatamente no ponto $R = P + Q$, o resultado da adição.

Como já foi excluído $Q = \pm P$ e $Q = \mathcal{O}$ da adição, então há três casos especiais:

1. Dado $Q = P$. A linha l_1 secante a P e Q não é mais a única. Em vez disso, pode-se escolher a linha l_1 como a tangente da curva que passa pelo ponto P.

2. Dado $Q = -P$. A linha l_1 não intercepta mais a curva. Define-se o ponto resultante R como o ponto no infinito $\mathcal{O}$.

3. Se $Q = \mathcal{O}$, então tem-se $P + \mathcal{O} = P$, pois $\mathcal{O}$ é o elemento neutro.

[3] tradução do inglês *distortion map*

[4] o endomorfismo não-racional é uma função racional

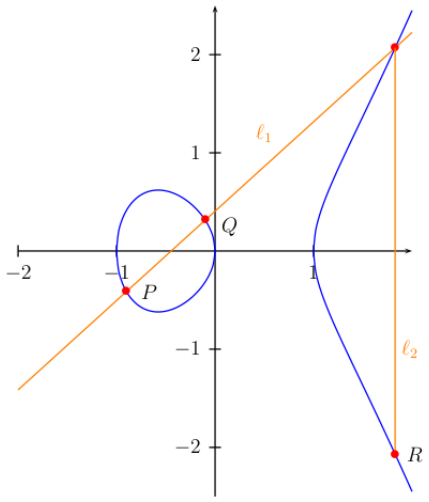


Figura 3.2: *Exemplos das operações de curvas elípticas*

O conjunto de pontos de uma curva elíptica sobre um corpo finito com o ponto no infinito $\mathcal{O}$ como um elemento neutro e a adição de pontos formam um grupo abeliano.

Por exemplo, para a curva $E/\mathbb{F}_5 : y^2 = x^3 + 1$ tem-se os pontos:

Ponto	(x, y)
$\hat{P}_1$	$(0, 1)$
$\hat{P}_2$	$(0, 4)$
$\hat{P}_3$	$(2, 2)$
$\hat{P}_4$	$(2, 3)$
$\hat{P}_5$	$(4, 0)$

Tabela 3.3: *Pontos na curva* $E/\mathbb{F}_5 : y^2 = x^3 + 1$

As operações do grupo em relação à adição da curva $E/\mathbb{F}_5$ é então

$+$	$\mathcal{O}$	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	$\hat{P}_4$	$\hat{P}_5$
$\mathcal{O}$	$\mathcal{O}$	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	$\hat{P}_4$	$\hat{P}_5$
$\hat{P}_1$	$\hat{P}_1$	$\hat{P}_2$	$\mathcal{O}$	$\hat{P}_4$	$\hat{P}_5$	$\hat{P}_3$
$\hat{P}_2$	$\hat{P}_2$	$\mathcal{O}$	$\hat{P}_1$	$\hat{P}_5$	$\hat{P}_3$	$\hat{P}_4$
$\hat{P}_3$	$\hat{P}_3$	$\hat{P}_4$	$\hat{P}_5$	$\hat{P}_2$	$\mathcal{O}$	$\hat{P}_1$
$\hat{P}_4$	$\hat{P}_4$	$\hat{P}_5$	$\hat{P}_3$	$\mathcal{O}$	$\hat{P}_1$	$\hat{P}_2$
$\hat{P}_5$	$\hat{P}_5$	$\hat{P}_3$	$\hat{P}_4$	$\hat{P}_1$	$\hat{P}_2$	$\mathcal{O}$

Tabela 3.4: *Adição de pontos na curva* $y^2 = x^3 + 1$ *sobre* $\mathbb{F}_5$

3.5 Coordenadas Projetivas

Lidar com o ponto no infinito $\mathcal{O}$ em uma curva elíptica, pode ser problemático usando as coordenadas afim, as coordenadas habituais (x, y) usadas para definir a forma normal de Weierstraßde uma curva elíptica.

Um artifício para lidar com essa questão é através do uso de coordenadas projetivas. As coordenadas projetivas codificam um ponto (x, y) com duas coordenadas em três coordenadas (X, Y, Z),

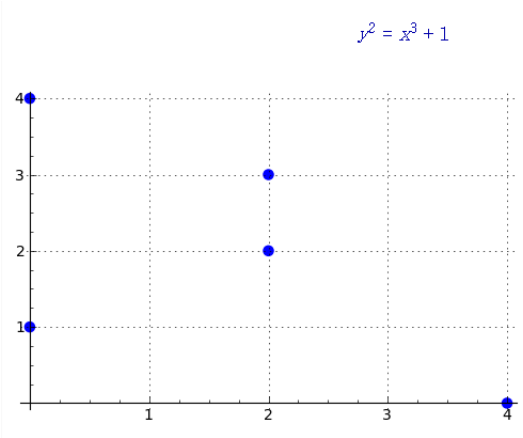


Figura 3.3: *Pontos da curva elíptica sobre* $\mathbb{F}_5$

onde (X, Y, Z) representa qualquer ponto da forma $(X/Z, Y/Z)$. Essas coordenadas projetivas são chamados de *coordenadas projetivas padrão*.

Em particular, representa-se um ponto em uma curva elíptica $P = (x, y)$ como $(x, y, 1)$ e o ponto no infinito pode ser representado por $(0, 1, 0)$. Também é possível converter facilmente coordenadas projetivas (X, Y, Z) onde $Z \neq 0$ em coordenadas afins $(X/Z, Y/Z)$.

Além de ser um bom método para lidar com o ponto no infinito, as coordenadas projetivas são frequentemente úteis na realização de cálculos em curvas elípticas.

Pois é possível adicionar dois pontos de uma curva elíptica que usam coordenadas projetivas sem executar nenhuma divisão, que é tipicamente muito cara computacionalmente em corpos finitos.

Finalmente, pelo fato de diferentes valores de Z serem usados para representar o mesmo ponto afim (x, y), então é possível utilizar valores aleatórios de Z para codificar tais pontos, isso irá fornecer um nível adicional de proteção contra os ataques por canais secundários.

Os ataques a uma implementação de um algoritmo criptográfico que buscam encontrar informações sobre a chave utilizada, através de medições físicas de um dispositivo em funcionamento e seu ambiente.

Em aplicações criptográficas, onde deseja-se executar operações em $\mathbb{F}_q$ para valores grandes de q, determinar o inverso de um elemento de $\mathbb{F}_q$ pode ser bastante caro em relação à multiplicação em $\mathbb{F}_q$, e o uso de coordenadas projetivas fornece uma vantagem de desempenho sobre o uso de coordenadas afim.

O sistema de coordenadas projetivas é representado por $\mathbb{P}^2$.

Observe que há outras formas de coordenadas projetivas que podem ser úteis para a aritmética de curvas elípticas. Estas formas de coordenadas projetivas requerem procedimentos diferentes para a adição de pontos, as técnicas apresentadas abaixo.

Particularmente, as coordenadas projetivas Jacobianas codificam um ponto afim $(X/Z^2, Y/Z^3)$ como o ponto projetivo (X, Y, Z) (Washington, 2008, sec. 2.6).

3.5.1 Operações de pontos em coordenadas projetivas

Cada tipo de coordenadas projetivas requer um número diferente de operações de corpo para somar ou duplicar pontos, que é resumida na Tabela 3.5, pág.23. A escolha do sistema de coordenada

projetiva mais eficiente vai depender da aplicação[5].

Em coordenadas projetivas padrão, dados pontos $P_1 = (X_1, Y_1, Z_1)$ e $P_2 = (X_2, Y_2, Z_2)$, deseja-se encontrar o ponto $P_3 = (X_3, Y_3, Z_3)$, tal que, $P_3 = P_1 + P_2$. Então, podem-se converter as coordenadas projetivas para coordenadas afim, onde $Q_1 = (X_1/Z_1, Y_1/Z_1)$ e $Q_2 = (X_2/Z_2, Y_2/Z_2)$

Desse modo, a adição de coordenadas projetivas padrão é:

$$\begin{aligned}
U_1 &= X_1 * Z_2 \\
U_2 &= X_2 * Z_1 \\
S_1 &= Y_1 * Z_2 \\
S_2 &= Y_2 * Z_1 \\
W &= Z_1 * Z_2 \\
P &= U_2 - U_1 \\
R &= S_2 - S_1 \\
X_3 &= P * (-(U_1 + U_2) * P^2 + W * R^2) \\
Y_3 &= (R * (-2 * W * R^2 + 3 * (U_1 + U_2) * P^2) - P^3 * (S_1 + S_2))/2 \\
Z_3 &= W * P^3
\end{aligned}$$

A fórmula para duplicação de pontos em coordenadas projetivas padrão é:

$$\begin{aligned}
w &= a * Z_1^2 + 3 * X_1^2 \\
s &= Y_1 * Z_1 \\
B &= X_1 * Y_1 * s \\
h &= w^2 - 8 * B \\
X3 &= 2 * h * s \\
Y3 &= w * (4 * B - h) - 8 * Y_1^2 * s^2 \\
Z3 &= 8 * s^3
\end{aligned}$$

Para o sistema de coordenadas Jacobianas, dados pontos $P_1 = (X_1, Y_1, Z_1)$ e $P_2 = (X_2, Y_2, Z_2)$, deseja-se encontrar o ponto $P_3 = (X_3, Y_3, Z_3)$, tal que, $P_3 = P_1 + P_2$. Então, podem-se converter as coordenadas projetivas para coordenadas afim, onde $Q_1 = (X_1/Z_1^2, Y_1/Z_1^3)$ e $Q_2 = (X_2/Z_2^2, Y_2/Z_2^3)$

A adição de coordenadas Jacobianas é:

$$\begin{aligned}
U_1 &= X_1 * Z_2^2 \\
U_2 &= X_2 * Z_1^2 \\
S_1 &= Y_1 * Z_2^3 \\
S_2 &= Y_2 * Z_1^3 \\
P &= U_2 - U_1 \\
R &= S_2 - S_1 \\
X_3 &= -(U_1 + U_2) * P^2 + R^2 \\
Y_3 &= -S1 * P^3 + R * (U_1 * P^2 - X_3) \\
Z_3 &= Z_1 * Z_2 * P
\end{aligned}$$

A fórmula para duplicação de pontos em coordenadas Jacobianas é:

[5] As fórmulas otimizadas foram retiradas de http://www.hyperelliptic.org/EFD/

$$\begin{aligned} S &= 4 * X1 * Y1^2 \\ M &= 3 * X1^2 + a * Z1^4 \\ T &= -2 * S + M^2 \\ X3 &= T \\ Y3 &= -8 * Y1^4 + M * (S - T) \\ Z3 &= 2 * Y1 * Z1 \end{aligned}$$

Sistema de coordenadas	Adição de Pontos	Duplicação de Pontos
Afim	I+2M+2Q	I+2M+1Q
Jacobiana	12M+4Q	4M+6Q
Padrão	12M+2Q	7M+5Q

Tabela 3.5: *As operações de corpo necessárias para implementar operações em curva elípticas em sistemas diferentes de coordenadas.*

Onde n multiplicações de corpo e m quadrados de corpo é indicado pela notação $nM + mQ$ e I indica que uma inversão também é necessária.

Se apenas adições de pontos são necessárias, a coordenada mais eficiente é usar coordenadas projetivas padrão.

Se só são necessárias duplicações pontos, a coordenada mais eficiente é usar coordenada projetiva Jacobiana.

Na maioria dos casos, é mais eficiente usar coordenada projetiva Jacobiana. Operações de duplicação de pontos podem ser otimizadas se o coeficiente $a = -3$ na forma normal de Weierstraß de uma curva elíptica.

Tabela 3.6: *Comparação entre afins e projetivas*

Operações	C. Afim (ms)	C. Projetiva (ms)	Eficiência (em %)
Soma de pontos	35	21	40%
Multiplicação de P por escalar	57	35	38,6%

O resultado obtido na utilização, da biblioteca implementada em Java, das transformações de coordenadas afins para coordenadas projetivas confirma nossas expectativas.

3.6 Estruturas algébricas de curvas elípticas

Os pontos de uma curva elíptica fornecem uma estrutura que pode ser definida em terminologia da álgebra.

Definição 3.5. Se E é uma curva elíptica sobre um corpo $\mathbb{F}$, então $E(\mathbb{F})$ indica o conjunto de pontos de E juntamente com a operação de adição de pontos.

Propriedade 3.3. *(Martin, 2008, prop.3.2)* Se $\mathbb{F}$ é um corpo e E é uma curva elíptica, então $E(\mathbb{F})$ é um grupo.

O ponto no infinito $\mathcal{O}$ atua como elemento identidade para este grupo.

Note que existe apenas uma operação definida para $E(\mathbb{F})$, que é sugerido como a adição, por isso é impossível multiplicar ou dividir os elementos de $E(\mathbb{F})$.

Assim, $E(\mathbb{F})$ não pode ser um corpo que requer duas operações, por exemplo, a adição e a multiplicação.

Definição 3.6. Multiplicação de um ponto P em uma curva elíptica por um inteiro n é o resultado da adição de um ponto por si mesmo n vezes, de forma que

$$nP = \underbrace{P + P + \ldots + P}_{n \text{ vezes}}$$

Definição 3.7. Seja $P \in E(\mathbb{F})$ para alguma curva elíptica $E/\mathbb{F}$. Diz-se que a *ordem* de um ponto é n se n é o menor inteiro positivo tal que $nP = \mathcal{O}$.

Definição 3.8. *(Martin, 2008, def.3.8)* Se E é uma curva elíptica sobre um corpo $\mathbb{F}$ e n é um inteiro positivo, escreve-se $E(\mathbb{F})[n]$ para o conjunto dos pontos de ordem n em $E(\mathbb{F})$.

Se o corpo $\mathbb{F}$ está bem definido, este pode ser abreviado para $E[n]$. $E(\mathbb{F})[n]$ é um subgrupo de $E(\mathbb{F})$.

Os pontos em $E(\mathbb{F})[n]$ também são chamados de *pontos de n-torção* da curva E.

3.6.1 Ordem da curva elíptica

Definição 3.9. Escreve-se $\#E(\mathbb{F})$ para indicar a ordem do grupo $E(\mathbb{F})$, que é o número de pontos na curva elíptica E sobre o corpo $\mathbb{F}$, incluindo o ponto no infinito, $\mathcal{O}$.

Determinar o valor de $\#E(\mathbb{F})$ para uma curva elíptica arbitrária é um problema não trivial.

Definição 3.10. Se E é uma curva elíptica sobre $\mathbb{F}_q$ e temos $\#E(\mathbb{F}_q) = q+1-t$, então t é chamado de *traço de Frobenius*, ou simplesmente o *traço.*

Espera-se ter cerca de $q+1$ pontos sobre uma curva elíptica $E/\mathbb{F}_q$.

A equação $y^2 = x^3 + ax + b$ tem uma solução quando $x^3 + ax + b$ é um resíduo quadrático módulo q, que deverá acontecer cerca de metade das vezes.

Em cada um destes casos, pode-se obter um par de raízes quadradas, assim é esperado que uma curva elíptica aleatória tenha cerca de q pontos finitos mais o ponto no infinito, para um total de $q+1$ pontos.

O Teorema de Hasse diz que uma curva elíptica $E/\mathbb{F}_q$ tem que ter aproximadamente $q+1$ pontos sobre ela, e que o traço diz aproximadamente quão longe deste comportamento esperado uma curva particular está.

Propriedade 3.4 (Teorema de Hasse)**.** *(Martin, 2008, prop.3.3)* Para uma curva elíptica $E/\mathbb{F}_q$, *o traço de Frobenius* satisfaz a desigualdade $|t| \leq 2\sqrt{q}$. Assim, o número de pontos em uma curva elíptica sobre q é aproximadamente $q+1$.

Definição 3.11. Se E é uma curva elíptica sobre q e $\#E(\mathbb{F}_q) = q$, então diz-se que E é *anômala.*

A descrição do algoritmo, apresentado em Blake *et al.* (2005) e Silverman (1986) (prep.6.5 da sec.XI.6), usado para calcular eficientemente o logaritmo discreto em curvas anômalas está além do escopo desta dissertação.

Sabe-se que este algoritmo executa em tempo linear, fazendo de tais curvas inadequadas para o uso na maioria das aplicações de criptografia.

3.6.2 Tipos de curvas

Definição 3.12. Seja p a característica de $\mathbb{F}_q$, E é uma curva elíptica sobre $\mathbb{F}_q$ e t o traço de E. Se p divide t, então diz-se que a curva elíptica E é *supersingular*.

Uma curva que não é supersingular é dita *ordinária*.

Note que os conceitos de singular e supersingular são muito diferentes e não devem ser confundidos com os conceitos apresentados na Seção 3.2.

Propriedade 3.5. Se $E : y^2 = f(x)$ é uma curva elíptica sobre $\mathbb{F}_q$ então E é *supersingular* exatamente quando o coeficiente de x^{p-1} em

$$(f(x))^{\frac{p-1}{2}}$$

é zero *(Silverman, 1986, sec.V.4 pág.141)*.

3.6.3 Grupo multiplicativo

Os pontos em uma curva elíptica formam um grupo, mas é necessária a estrutura de um corpo para executar os cálculos que alguns algoritmos de protocolos sem certificados necessitam.

Para fazer isso, insere-se um grupo de curva elíptica em um corpo finito.

Em muitos casos, isso irá resultar em um corpo finito que é grande demais para ter cálculos práticos.

Definição 3.13. Seja $E/\mathbb{F}_q$ uma curva elíptica e n um número inteiro tal que $n|\#E(\mathbb{F}_q)$.

Se k é o menor inteiro positivo tal que $n|(q^k - 1)$ então k é chamado de *grau de mergulho* de E em relação a n.

Se $n = \#E(\mathbb{F}_q)$, então pode-se abreviar isto dizendo que k é o grau de mergulho de E.

Se k é o grau de mergulho de $E/\mathbb{F}_q$, pode-se pensar $\mathbb{F}_{q^k}$ como sendo uma extensão de $\mathbb{F}_q$ em que $E(\mathbb{F}_q)$ é um subgrupo de $\mathbb{F}^*_{q^k}$.

Isso permite a multiplicação de pontos, uma operação que não é possível realizar em um grupo de curva elíptica onde somente a operação de adição é definida.

Um pequeno grau de mergulho faz que alguns algoritmos de criptografia de curva elíptica sejam vulneráveis a alguns ataques de criptoanálise, então é necessário selecionar os parâmetros dos algoritmos que usam essas curvas com cuidado para evitar tais fraquezas.

Definição 3.14. Se $E/\mathbb{F}_q$ é uma curva supersingular com $q = p^n$ e traço t, então a Tabela 3.7 lista as possíveis classes de curvas supersingulares *(Darrel Hankerson, 2004, sec.4.1 pág.169)*.

Em particular, alguma curva supersingular tem grau de mergulho $k = 6$, e para $E/\mathbb{F}_q$ com $q > 3$, os três únicos casos possíveis são $n = 1$, $n = 2$ e $n = 3$.

3.7 Mais alguns conceitos

Mais alguns conceitos são necessários para compreender as curvas utilizadas em criptossistemas baseados em emparelhamentos.

Classe	Traço t	Grau de mergulho k	Parâmetros
1	0	2	$E(\mathbb{F}_q) \equiv \mathbb{Z}_{q+1}$ e p primo
2	0	2	$E(\mathbb{F}_q) \equiv \mathbb{Z}_{(q+1)/2}$ e $q \equiv 3 \pmod 4$
3	q	3	p primo e n par
4	$2q$	4	$p = 2$, n ímpar
5	$3q$	6	$p = 3$, n ímpar
6	$4q$	1	p primo e n par

Tabela 3.7: *Classificação das curvas supersingulares*

3.7.1 Multiplicação complexa

Todos os grupos de curvas elípticas têm alguns endomorfismos: as aplicações de multiplicação por n (escalar) da forma $f_n(P) = nP$. Alguns grupos de curvas elípticas têm endomorfismos adicionais que não são isomorfos a tal aplicação de multiplicação pelo escalar n (Martin, 2008, sec.3.3.2 pág.65).

Uma curva elíptica com tal propriedade é dita possuir a multiplicação complexa, que pode-se abreviar como "MC".

O termo "*multiplicação complexa*" vem do fato que em muitos casos, estes endomorfismos agem como multiplicação por um número complexo.

Assim, pode-se ter que $f(f(P)) = -D \cdot P$ para algum $D > 0$, de modo que $f \circ f = -D$ ou $f^2 = -D$, sugere que f atue como multiplicando pelo número imaginário $\sqrt{-D}$.

Veremos em capítulos posteriores que existem técnicas que funcionam em curvas com a multiplicação complexa que pode ser usada para gerar curvas elípticas adequadas para emparelhamentos amigáveis.

3.7.2 Raízes n-ésimas da unidade

O que define a imagem da aplicação bilinear denominado de *raízes n-ésimas da unidade*, ou número de *De Moivre*.

Definição 3.15. Seja

$$\mu_n = \{x \in \bar{k} | x^n = 1\}$$

o grupo das raízes n-ésimas da unidade. Desde que n é co-primo à característica do corpo k, $\#\mu_n = n$, isto é, μ_n é um grupo cíclico de ordem n. (Jeffrey Hoffstein, 2008, sec.5.8.5 pág.325)

3.7.3 Polinômios ciclotômicos

Como descrito na Seção 4.2, onde é necessária a exponenciação no corpo de extensão com uma forma específica, necessita-se do conceito de subgrupo ciclotômico. Este, por sua vez, utiliza-se da definição do polinômio ciclotômico.

Definição 3.16. Um *subgrupo ciclotômico* de corpo de extensão amigável à torre $\mathbb{F}_{p^k}$, com $k = 2^a 3^b$, $a, b \geq 1$ é um subgrupo de ordem $\Phi_k(p)$.

Definição 3.17. O polinômio Φ_k é o k-ésimo *polinômio ciclotômico*, que quando $6|k$, para o k descrito, é sempre da forma

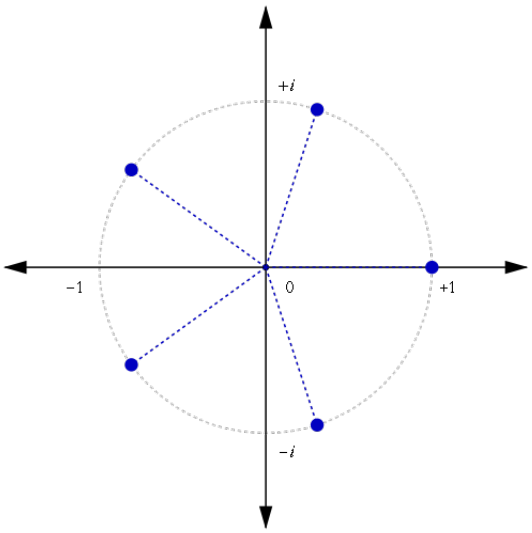


Figura 3.4: *Exemplos do S^1 com as raízes n-ésimas da unidade*

$$\Phi_{2^a3^b}(x) = x^{2\cdot 2^{a-1}3^{b-1}} - x^{2^{a-1}3^{b-1}} + 1$$

denota-se o subgrupo ciclotômco por $\mathbb{G}_{\Phi_k(p)}$, sendo o conjunto formado da seguinte forma

$$\mathbb{G}_{\Phi_k(p)} = \{\alpha \in \mathbb{F}_{p^k} | \alpha^{\Phi_k(p)} = 1\}$$

Essas definições de raízes da unidade e de subgrupo ciclotômico serão muito importantes para interpretação do resultado do algoritmo de Miller, e para compreensão dos tipos de emparelhamentos no próximo capítulo.

3.8 Criptografia de Curvas Elípticas

A Criptografia de Curvas Elípticas, ou ECC, das iniciais em inglês "*Elliptic Curve Cryptography*", é uma variante da criptografia assimétrica ou de chave pública, baseada na matemática das curvas elípticas. Seus criadores argumentam que a ECC pode ser mais rápida e usar chaves mais curtas do que os métodos antigos - como RSA -, e proporcionar ao mesmo tempo um nível de segurança equivalente. A utilização de curvas elípticas em criptografia foi proposta de modo independente por Neal Koblitz e Victor Miller em 1985 (Koblitz, 1987).

Existem várias curvas elípticas utilizadas em criptografia distintas por pequenas variações, mas todas elas se baseiam na dificuldade de se resolver o problema do logaritmo discreto e os problemas relacionados para o grupo de uma curva elíptica sobre grupos finitos. Os grupos finitos mais usados são os inteiros módulo um número primo, ou um grupo de Galois cujo tamanho é potência de 2.

Dada uma curva elíptica E, e um grupo $GF(q)$, consideramos o grupo abeliano de pontos racionais $E(q)$ na forma (x, y), onde x e y pertencem a $GF(q)$, e onde a operação de grupo $+$ se define nesta curva.

Define-se então uma segunda operação "$\star$" $|\mathbb{Z} \times E(q) \rightarrow E(q)$: se P é algum ponto em $E(q)$, então definimos

$$\begin{cases} 2 \star P &= P + P & , \\ 3 \star P &= 2 \star P + P &= P + P + P, \end{cases}$$

e assim por diante (Miller, 1986).

Nota-se que dados os inteiros j e k, $j \star (k \star P) = (j \star k) \star P = k \star (j \star P)$. O problema do logaritmo discreto de uma curva elíptica (PLDCE) é dados os pontos P e Q determinar o inteiro k quando $k \star P = Q$ (Ian F. Blake, 1999).

Acredita-se que o problema do logaritmo discreto sobre o grupo multiplicativo (DLP) e o PLDCE não são problemas equivalentes; ao contrário, crê-se que o PLDCE é significativamente mais difícil do que o DLP (Al-Riyami e Paterson, 2003).

Em criptografia, escolhe-se um ponto base Q específico e público para se usar com a curva $E(q)$. Escolhe-se um número inteiro aleatório k como chave secreta. Então o valor $P = k \star Q$ é divulgado como chave pública. Note que a suposta dificuldade da PLDCE implica que é difícil deduzir k a partir de P.

Se Alice e Beto têm as chaves particulares k_A e k_B, então Alice poderia calcular $k_A \star P_B = (k_A \star k_B) \star Q$. E Beto pode obter o mesmo valor dado que $k_B \star P_A = (k_B \star k_A) \star Q$.

Isto permite estabelecer um "valor secreto" que tanto Alice quanto Beto podem calcular facilmente, por meio da estrutura bilinear, mas que é muito mais complicado de ser derivado por uma terceira pessoa. Além disso, Beto não consegue averiguar nada novo sobre k_A, de modo que a chave de Alice continua a ser secreta.

3.9 Resumo

Neste capítulo, apresentamos os conceitos preliminares sobre curvas elípticas para compreensão do próximo capítulo sobre emparelhamentos bilineares.

Dentre eles, foram apresentadas noções sobre curvas elípticas e a aritmética envolvida, torções em curvas, mudança de coordenadas dos pontos na curva, estrutura algébrica de uma curva.

E outras operações envolvidas nos emparelhamentos como: a multiplicação complexa e as raízes da unidade.

Capítulo 4

Emparelhamento bilinear

Neste capítulo, considerando as definições apresentadas no Apêndice B sobre divisores, apresentaremos uma introdução aos emparelhamentos, e seus tipos conforme a utilização da curva elíptica adequada.

Lembrando das definições do capítulo anterior; Seja $\mathcal{G} = E[q]$, os pontos de ordem q de uma curva elíptica.

Assume-se que a curva definida sobre um corpo finito $\mathbb{F}_p$ e que o grupo $E[q]$ está contido em $E(\mathbb{F}_{p^k})$, onde por simplicidade e eficiência, assume-se que k é par.

O grupo $\mathcal{G}$ é um produto de dois grupos cíclicos $\mathcal{G}_1$ e $\mathcal{G}_2$ de ordem q.

Toma-se um ponto $\mathcal{P}_1 \in E(\mathbb{F}_p)$ como um gerador de $\mathcal{G}_1$ e um ponto $\mathcal{P}_2 \in E(\mathbb{F}_{p^k})$ como um gerador de $\mathcal{G}_2$.

Escolhe-se $\mathcal{P}_2$ de modo que ele esteja na imagem da torção[1] quadrática de E sobre $\mathbb{F}_{p^{k/2}}$.

Pode-se definir então, um emparelhamento e de $\mathcal{G} \times \mathcal{G}$ para o subgrupo $\mathcal{G}_T$ de ordem q do corpo finito $\mathbb{F}_{p^k}$.

Definição 4.1. Um *emparelhamento* é uma aplicação bilinear $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ entre os três grupos $\mathbb{G}_1$, $\mathbb{G}_2$ e $\mathbb{G}_T$ de expoente q, que tem as seguintes propriedades:

Bilinear Diz-se que uma aplicação $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ é *bilinear* se $e(aP, bQ) = e(P, Q)^{ab}$ para todo $P \in \mathbb{G}_1$, todo $Q \in \mathbb{G}_2$ e $a, b \in \mathbb{Z}_q$.

Não degenerado diz-se que e é *não degenerado* se esta aplicação não enviar nenhum dos pares em $\mathbb{G}_1 \times \mathbb{G}_2$ para o elemento identidade em $\mathbb{G}_T$. Desde que $\mathbb{G}_1$ e $\mathbb{G}_2$ são grupos de ordem q primo, assim se $P \in \mathbb{G}_1$ é um gerador de $\mathbb{G}_1$, e $Q \in \mathbb{G}_2$ é um gerador de $\mathbb{G}_2$ então $e(P, Q)$ é um gerador de $\mathbb{G}_T$.

Computável Há um algoritmo eficiente para calcular $e(P, Q)$ para qualquer $P \in \mathbb{G}_1$ e $Q \in \mathbb{G}_2$.

Este emparelhamento é trivial se e somente se os dois valores de entrada forem linearmente dependentes no espaço vetorial $E[q]$.

Será conveniente definir quatro tipos diferentes de sistemas de emparelhamento, três das quais são tomadas a partir de Galbraith *et al.* (2006) enquanto o quarto tipo é citado em Shacham (2005).

[1] tradução do inglês twist

Além disso, é esperado que seja fácil produzir uma função de hash criptográfico que calcule "*hashs*" para $\mathcal{G}_1$, $\mathcal{G}_2$ ou $\mathcal{G}$, mas que não seja fácil produzir uma função que calcule "*hashs*" para qualquer outro subgrupo de ordem q de $\mathcal{G}$, $\mathcal{G}_1$ e $\mathcal{G}_2$ (Chen *et al.*, 2007, sec.2 pág.3).

Utilizar todo o grupo $\mathcal{G}$ para as coordenadas em um sistema criptográfico baseado em emparelhamento é muito ineficiente.

Assim, na literatura existe uma série de especializações da situação acima. Dentre elas, as quatro mais importantes serão apresentadas a seguir. Em todos os casos nós temos $\mathbb{G}_T = \mathcal{G}_T$.

Definição 4.2 (Emparelhamento Tipo 1)**.** *(Chen et al., 2007, def.2 pág.4)* Define-se um emparelhamento utilizando a chamada *função de distorção*[2], sendo $\mathbb{G}_1 = \mathbb{G}_2 = \mathcal{G}_1$, temos $P_1 = P_2 = \mathcal{P}_1$. Tal situação corresponde aos denominados emparelhamentos simétricos sobre curvas elípticas supersingulares. Há um algoritmo de hash criptográfico eficiente para cadeias de bits arbitrárias em $\mathbb{G}_1$ e em $\mathbb{G}_2$ e um isomorfismo (trivial) de grupo $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ aplicando P_2 em P_1. ■

Os três tipos seguintes correspondem aos *emparelhamentos assimétricos* sobre curvas elípticas *ordinárias* ou não supersingulares. Deste modo, a necessidade da existência do homomorfismo de $\mathbb{G}_2 \to \mathbb{G}_1$ será explicado na Seção 4.4, quando define-se emparelhamento Ate.

Definição 4.3 (Emparelhamento Tipo 2)**.** *(Chen et al., 2007, def.3)* Nesta situação, toma-se $\mathbb{G}_1 = \mathcal{G}_1$ e $\mathbb{G}_2$ é um subgrupo de $\mathcal{G}$ que não é igual a $\mathcal{G}_1$ nem a $\mathcal{G}_2$. Define-se $P_1 = \mathcal{P}_1$ e por conveniência fixa-se $P_2 = \frac{1}{k}\mathcal{P}_1 + \mathcal{P}_2$. Há um algoritmo de hash criptográfico eficiente para cadeias de bits arbitrárias em $\mathbb{G}_1$, mas acredita-se que não exista um método para calcular cadeias de bits em $\mathbb{G}_2$. ■

No entanto, há um isomorfismo de grupo ψ eficientemente computável: $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ aplicando P_2 em P_1, que é a função traço restrita a $\mathbb{G}_2$.

Definição 4.4 (Emparelhamento Tipo 3)**.** *(Chen et al., 2007, def.4)* Nesta situação, tem-se $\mathbb{G}_1 = \mathcal{G}_1$ e $\mathbb{G}_2 = \mathcal{G}_2$, com geradores de $P_1 = \mathcal{P}_1$ e $P_2 = \mathcal{P}_2$. Há um algoritmo de hash criptográfico eficiente para cadeias de bits arbitrárias em $\mathbb{G}_1$, e um algoritmo de hash criptográfico um pouco menos eficiente para cadeias de bits aleatórias em $\mathbb{G}_2$. ■

Apesar disso, não há conhecimento de uma forma eficiente de isomorfismo de grupo computável $\phi : \mathbb{G}_2 \to \mathbb{G}_1$ aplicado P_2 a P_1.

Definição 4.5 (Emparelhamento Tipo 4)**.** *(Chen et al., 2007, def.5 pág.5)* Toma-se $\mathbb{G}_1 = \mathcal{G}_1$, mas define-se $\mathbb{G}_2$ como todo o grupo $\mathbb{G}$, que tem ordem q^2.Como no Tipo 2, fixa-se $P_1 = \mathcal{P}_1$ e $P_2 = \frac{1}{k}\mathcal{P}_1 + \mathcal{P}_2$. Há um homomorfismo eficientemente computável ψ de G_2 em G_1 tal que $\psi(P_2) = P_1$. ■

Pode-se executar a função de hash em G_1 e em G_2, porém em G_2 não seja muito eficiente.

Contudo, não deve existir uma função de hash eficiente no subgrupo de G_2 gerado por P_2. Note que o emparelhamento de um elemento diferente do zero de G_1 em G_2 pode ser trivial nessa situação.

Assim, em todas as situações temos que P_1 é o gerador do $\mathbb{G}_1$ e P_2 é um elemento fixo de $\mathbb{G}_2$ da ordem prima q, de modo que onde existe um homomorfismo computável ψ de $\mathbb{G}_2$ para $\mathbb{G}_1$ temos $\psi(P_2) = P_1$.

Em emparelhamentos do Tipo 3, se tal isomorfismo existir, acredita-se que não será possível calculá-lo, mesmo definindo-se ψ nesta situação, mas deve-se ter em mente que será incapaz de calculá-lo.

[2] tradução do inglês de *distortion maps*

Chen *et al.* (2007) (tab.1 pág.11) apresentaram uma série de protocolos eficientes de acordo de chave que podem ser implementados no cenário do Tipo 3, ou menos eficiente na configuração do Tipo 2. Os demais protocolos são implementáveis apenas na configuração do Tipo 1 e Tipo 4.

Na configuração do Tipo 1, tem-se problemas devido à eficiência, como o parâmetro de segurança aumenta à medida que são restritos a curvas supersingulares.

Na configuração do Tipo 4, as provas de segurança tornam-se mais complicadas, pois a imagem da função de hash em $\mathbb{G}_2$ não estará no grupo gerado por P_2.

Deve-se referir aos grupos $\mathbb{G}_1$, $\mathbb{G}_2$ e $\mathbb{G}_T$, os elementos P_1 e P_2, o emparelhamento e, e possivelmente o homomorfismo ψ, como um conjunto de parâmetros do emparelhamento.

Assume-se que, dado um parâmetro de segurança pode-se gerar um conjunto de parâmetros de emparelhamento associado ao nível de segurança necessário.

Em um artigo recente, Chatterjee e Menezes (2009) apresentam um método eficiente para converter o emparelhamento do Tipo 2 em Tipo 3, e argumentam que ao contrário do que se acreditava, os emparelhamentos do Tipo 2 são apenas implementações ineficientes de emparelhamentos do Tipo 3, e parecem oferecer nenhum benefício para protocolos baseados em emparelhamentos assimétricos do ponto de vista de funcionalidade, segurança e desempenho. Dessa forma, os únicos emparelhamentos relevantes são Tipo 1 que é simétrico, e os Tipo 3 e 4 assimétricos.

4.1 Emparelhamento de Weil

O emparelhamento de Weil foi inicialmente utilizado na demonstração da hipótese de Riemann para curvas.

Definição 4.6 (Emparelhamento de Weil, (Boneh e Franklin, 2001))**.** Seja E uma curva elíptica sobre $\mathbb{F}_q$, $r \in \mathbb{N}_0$, e tomando $m \in \mathbb{N}_0$ tal que $E[r] \subset E(\mathbb{F}_{q^m})$, então o *emparelhamento de Weil* w_r é uma aplicação

$$\begin{aligned} w_r : E[r] \times E[r] &\rightarrow \mu_r \subset \mathbb{F}_{q^m}; \\ (P, Q) &\mapsto w_r(P, Q) = \frac{f(D_Q)}{g(D_P)} \end{aligned}$$

onde $D_P \sim (P) - (\mathcal{O})$, $D_Q \sim (Q) - (\mathcal{O})$ com o suporte disjunto e $div(f) = rD_P$, $div(g) = rD_Q$.

Para provar a independência das escolhas feitas nos divisores D_P e D_Q.

Se D'_P é outro divisor linearmente equivalente com $(P) - (\mathcal{O})$, então, por definição, existe uma função h tal que $D'_P = D_P + div(h)$.

Se $f' = fh^r$, então $div(f') = rD'_P$ e, portanto

$$\frac{f'(D_Q)}{g(D'_P)} = \frac{f(D_Q)h^r(D_Q)}{g(D_P)g(div(h))} = \frac{f(D_Q)h^r(D_Q)}{g(D_P)h(div(g))} = \frac{f(D_Q)}{g(D_P)},$$

onde usa-se a reciprocidade de Weil, apresentado no Apêndice B.1, no denominador da segunda igualdade.

Um raciocínio semelhante (escolhe-se outro divisor $D'_Q = D_Q + div(k)$, $g' = gk^r$, $div(g') = rD'_Q$) demonstra a independência das escolhas feitas no D_Q.

Provar que a aplicação w_r em μ_r é igualmente simples:

$$w_r(P,Q)^r = \frac{f(rD_Q)}{g(rD_P)} = \frac{f(div(g))}{g(div(f))} = 1,$$

pela reciprocidade Weil.

Na definição do emparelhamento Weil, precisa-se de uma extensão de grau m tal que $E[n] \subset \mathbb{F}_{q^m}$.

Se r é um número primo grande, com $mdc(r,q) = 1$ e $r|\#E(\mathbb{F}_q)$, então m pode ser tomado como segue: se $k > 1$, então $m = k$ (o grau de mergulho), se $k = 1$ e $E[r] \subset E(\mathbb{F}_q)$, então $m = 1$, senão $m = r$.

Seja $P, Q \in E[r]$, então Q encontra-se no subgrupo gerado por P se e somente se $w_r(P,Q) = 1$.

Esta propriedade pode ser usada para determinar a estrutura de grupo de uma curva elíptica como descrito em Miller (2004) (sec. 6).

Na prática, trabalhar com a Definição 4.6 não é muito eficiente, pois precisamos valorar as funções f e g em cada dois pontos.

Lembre-se das funções Miller $f_{n,P}$ com divisor $div(f_{n,P}) = n(P) - ([n]P) - (n-1)(\mathcal{O})$, então temos a seguinte propriedade, a qual uma prova é fornecida em Miller (2004) no Teorema 2 e pág.241.

Propriedade 4.1. Seja E uma curva elíptica sobre $\mathbb{F}_q$ e seja $P, Q \in E[r]$ com $P \neq Q$, então

$$w_r(P,Q) = (-1)^r \frac{f_{r,P}(Q)}{f_{r,Q}(P)}.$$

4.1.1 Emparelhamento de Tate

O emparelhamento de Tate opera em pares de pontos $P \in E(\mathbb{F}_p)[n]$ e $Q \in E(\mathbb{F}_{p^k})$, e produz um resultado em $\mathbb{F}^*_{p^k}$. Escreve-se $e(P,Q)$ para o emparelhamento de Tate dos pontos P e Q.

Para um ponto P de ordem n obter $e(P,Q)$, primeiro encontra-se uma função racional f_P de modo que $div(f_P)$ é equivalente a $n(P) - n(\mathcal{O})$ e então valora-se f_P num divisor equivalente a $(Q) - (\mathcal{O})$. Isso pode ser resumido na seguinte definição.

Definição 4.7 (Definição ingênua)**.** Seja $E/\mathbb{F}_p$ uma curva elíptica, $P \in E(\mathbb{F}_p)[n]$ e $Q \in E(\mathbb{F}_{p^k})$. Seja f_P uma função racional com $div(f_P)$ equivalente a $n(P) - n(\mathcal{O})$ e A_Q sendo um divisor equivalente a $(Q) - (\mathcal{O})$ com o suporte do $div(f_P)$ e A_Q disjunto. Então o emparelhamento de Tate é definido sendo $e(P,Q) = f_P(A_Q)$.Esta definição não produz um único valor, e incluirá uma constante que é uma n-ésima potência de um elemento de $\mathbb{F}_{p^k}$.

Ao demonstrar que esta definição é realmente independente das escolhas para a função f_P e o divisor A_Q, observa-se que o emparelhamento de Tate é só definido salvo[3] a multiplicação por uma potência n-ésima de alguma constante (Martin, 2008, pág. 76).

A Seção 4.2 explicará como se livrar dessa constante indesejável, deixando um único valor.

Note que f_P é definido salvo um múltiplo constante. Aplicando a definição de valoração de um divisor em uma função para esse múltiplo constante, mostra que esta não tem qualquer influência sobre o valor de $f_P(A_Q)$, por isso é independente da escolha de f_P.

Mas antes, como foi feito para o emparelhamento de Weil, é necessário uma formalização mais precisa do emparelhamento de Tate.

[3] no sentido de "a menos de" ou "desconsiderando uma possível diferença de"

Definição 4.8. *(M. Joye, 2008, Definição II.15 pág.22)* Seja E uma curva elíptica sobre $\mathbb{F}_q$ e $r|\#E(\mathbb{F}_q)$, com $mdc(r,q)=1$, e seja k o grau de mergulho, então o emparelhamento de Tate $\hat{e}_r$ é a aplicação.

$$\begin{aligned}\hat{e}_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) &\rightarrow \mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^r \\ (P,Q) &\mapsto \hat{e}_r(P,Q) = f_{r,P}(D_Q),\end{aligned}$$

onde $div(f_r) = r(P) - r(\mathcal{O})$ e $D_Q \sim (Q) - (\mathcal{O})$ coprimo com $div(f_r)$.

Na definição acima, o ponto Q deve ser considerado como um representante de toda a classe de equivalência $Q + rE(\mathbb{F}^*_{q^k})$ e da mesma forma, o valor do emparelhamento de Tate é realmente a classe $f_{r,P}(D_Q)(\mathbb{F}^*_{q^k})^r$.

Como no emparelhamento de Weil, é necessário mostrar que a definição faz sentido, ou seja, é independente das escolhas feitas para o representante da classe $Q + rE(\mathbb{F}_{q^k})$ e o divisor D_Q.

Ambos seguem da reciprocidade de Weil da seguinte forma: para algum divisor $D'_Q = D_Q + div(h)$ co-primo com $div(f_r, P)$, temos:

$$f_{r,P}(D'_Q) = f_{r,P}(D_Q) f_{r,P}(div(h)),$$

e pela reciprocidade de Weil $f_{r,P}(div(h)) = h(r(P) - r(\mathcal{O})) = (h(P)/h(\mathcal{O}))^r \in (\mathbb{F}^*_{q^k})^r$.

Da mesma forma, substitui-se Q por $Q + rR$ para $R \in E(\mathbb{F}_{q^k})$, então

$$D_{Q+rR} \sim (Q + rR) - (\mathcal{O}) \sim (Q) + r(R) - (r-1)(\mathcal{O}) \sim D_Q + r(R) - r(\mathcal{O}),$$

onde a segunda equivalência decorre da definição da lei de grupo da curva elíptica.

Finalmente, isso implica $f_{r,P}(D_{Q+rR}) = f_{r,P}(D_Q) f((R) - (\mathcal{O}))^r$, o que prova a independência da escolha do representante.

Se r é primo, então pela definição do grau de mergulho temos $mdc(r, q^d - 1) = 1$ para todos os inteiros positivos $d|k$ e $d < k$.

Isso mostra que todos os elementos dos corpos $\mathbb{F}_{q^d}$ são potências r-ésimas.

Em particular, se $k > 1$ e se ambos P e Q são escolhidos para ter coordenadas em um subcorpo estrito de $\mathbb{F}_{q^k}$ então $\hat{e}_r(P,Q) = 1$, o que mostra que pelo menos um dos pontos de entrada foi definido sobre $\mathbb{F}_{q^k}$.

Na prática, frequentemente escolhe-se $P \in E(\mathbb{F}_q)[r]$ e Q necessariamente em $E(\mathbb{F}_{q^k}) \backslash \bigcup_{d|k, d<k} E(\mathbb{F}_{q^d})$.

O emparelhamento de Tate satisfaz propriedades semelhantes ao emparelhamento de Weil, como a bilinearidade, não é degenerado e é eficientemente computável.

Como para o emparelhamento de Weil, evita-se trabalhar com o divisor D_Q e simplesmente valorar em Q.

Para manter esta simplificação, necessita-se que a função $f_{r,P}$ seja devidamente normalizada.

Para compreender melhor o que isso significa.

Seja $u_{\mathcal{O}}$ um uniformizador $\mathbb{F}_q$-racional fixado em $\mathcal{O}$, então para qualquer função $f \in \bar{\mathbb{F}}_q(E)^*$ define-se $lc_{\mathcal{O}}(f)$ como o coeficiente líder de f. Sendo f uma série de Laurent em $u_{\mathcal{O}}$ (M. Joye, 2008, pág.23).

Note que quando f é definida em $\mathcal{O}$ simplesmente temos $f(\mathcal{O}) = lc_{\mathcal{O}}(f)$ independente do uniformizador escolhido.

Lema 4.1. Seja $P \in E(\mathbb{F}_{q^k})[r]$ e $Q \in E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ com $P \neq Q$, então

$$\hat{e}_r(P,Q) = f_{r,P}(Q) \cdot (\mathbb{F}_{q^k}^*)^r,$$

se e somente se $lc_{\mathcal{O}}(f_{r,P}) \in (\mathbb{F}_{q^k}^*)^r$.

Além disso, $lc_{\mathcal{O}}(f_{r,P})$ sendo uma r-ésima potência, é independente do uniformizador escolhido.

Esse lema mostra que, quando $f_{r,P}$ é devidamente normalizado, basta valorar em Q. A demonstração encontra-se em Granger *et al.* (2007) (Lema 1 pág.5).

Note que para $k > 1$ e $P \in E(\mathbb{F}_q)[r]$, qualquer função de Miller $\mathbb{F}_q$-racional $f_{r,P}$ será automaticamente normalizada desde que todos os elementos em $\mathbb{F}_q$ sejam r-ésimas potências, assim também $lc_{\mathcal{O}}(f_{r,P})$.

4.2 Exponenciações

Agora, suponha que D_1 e D_2 são divisores equivalentes a $(Q) - (\mathcal{O})$, diz-se $D_1 = D_2 + div(g)$ para alguma função racional g. É preciso ter cuidado, pois também será necessário assumir que o suporte da $div(f_P)$ é disjunto do suporte de $div(g)$. Então, tem-se que

$$\begin{aligned} f_P(D_1) &= f_P(D_2 + div(g)) \\ &= f_P(D_2) f_P(div(g)) \\ &= f_P(D_2) g(div(f_P)) \text{ (pela reciprocidade de Weil)} \\ &= f_P(D_2) g(r(P) - r(\mathcal{O})) \\ &= f_P(D_2) g((P) - (\mathcal{O}))^r \end{aligned}$$

Pode-se então abusar um pouco da notação de congruências para escrever isto como

$$f_P(D_1) \equiv f_P(D_2)$$

que foi pensado no sentido de que $f_P(D_1) = f_P(D_2)$ a menos de uma constante que é uma r-ésima potência.

Os exemplos de adição de divisores acima mostram como encontrar um divisor equivalente a $r(P) - r(\mathcal{O})$:

Pode-se adicionar o divisor $(P) - (\mathcal{O})$ com ele mesmo r vezes usando os divisores $div(u)$ e $div(v)$ que são obtidos a partir das linhas secantes através de vários pontos da curva elíptica.

Depois chegando a $r(P) - r(\mathcal{O})$ ficará com um divisor de uma função racional chamado f_P quando todos os termos que envolvem o ponto P anularam-se.

Para evitar problemas com a avaliação de uma função no ponto no infinito, que aparece em $(Q) - (\mathcal{O})$, como alternativa, escolhe-se um ponto aleatório R nessa curva elíptica e valora-se f_P em $(Q + R) - (R)$ que é equivalente ao divisor $(Q) - (\mathcal{O})$.

Porque o ponto P é de ordem r, se adicioná-lo repetidamente r vezes o divisor $(P) - (\mathcal{O})$ para obter $r(P) - r(\mathcal{O})$ usando a técnica descrita no Algoritmo 4.3 da pág.38.

Encontra-se como resultado um divisor de uma função racional que é o produto dos termos da forma u/v, onde u é a linha secante através de dois pontos (por exemplo, os pontos P_1 e P_2 na

Figura B.1, pág.88) nessa curva elíptica e v é a linha vertical que passa pelo ponto que é a soma dos mesmos dois pontos (o ponto P_3 na Figura B.1, por exemplo).

Suponha que A_Q é um divisor da forma $(Q+R)-(R)$ obtido de um $R \neq \mathcal{O}$ aleatório. Note que a exigência o qual o suporte dos divisores $r(P)-r(\mathcal{O})$ e A_Q sejam disjuntos significa que $Q+R \neq P$, e $R \neq P$.

Excluem-se estes casos, porque eles reduzem o valor do emparelhamento para zero pela inserção de um fator zero no cálculo, ou causam um erro de divisão por zero.

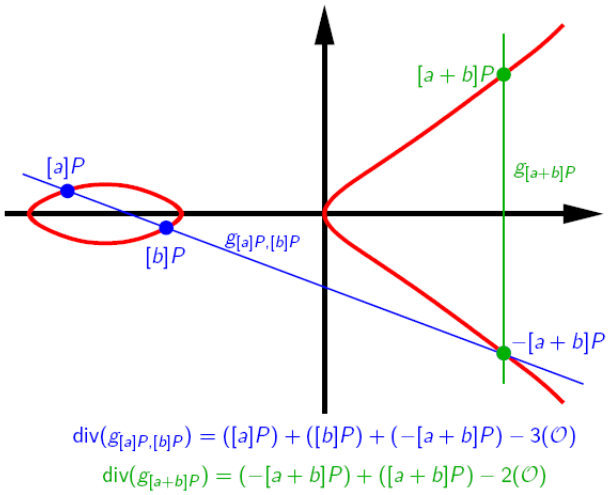


Figura 4.1: *Exemplo da utilização de divisores em operações de curvas elípticas*

Para dar um exemplo de funcionamento do emparelhamento, cita-se o mesmo exemplo usado na Seção 3.4 (pág.19) para encontrar $\hat{e}(\hat{P}_2, \hat{P}_2)$.

Exemplo 4.1. Descobre-se que $3(\hat{P}_2) - 3(\mathcal{O})$ é equivalente ao divisor $div(y+1)$, por isso tem-se $f_{\hat{P}_2} = y + 1$.

Em seguida, necessita-se de um ponto aleatório para adicionar $\hat{P}_2$, para o qual escolheu-se $\hat{P}_4$, por isso deseja-se valorar $f_{\hat{P}_2}$ em $(\hat{P}_2 + \hat{P}_4) - (\hat{P}_4) = (\hat{P}_3) - (\hat{P}_4)$, ou encontrar $f_{\hat{P}_2}(\hat{P}_3)/f_{\hat{P}_2}(\hat{P}_4)$.

Note que é possível escolher um ponto aleatório que provoca a divisão por zero, por exemplo, se o ponto $\hat{P}_2$ for escolhido neste exemplo. Se isso acontecer, pode-se apenas escolher outro ponto aleatório até encontrar um que funcione. Ao substituir os valores apropriados da Tabela 3.4 (pág.20), verifica-se que

$$\begin{aligned} e(\hat{P}_2, \hat{P}_2) &= \frac{f_{\hat{P}_2}(\hat{P}_3)}{f_{\hat{P}_2}(\hat{P}_4)} = \frac{3}{4} \\ &= 3.4^{-1} = 2 \in \mathbb{F}_5. \end{aligned} \tag{4.1}$$

Como mencionado acima, o emparelhamento de Tate possui um fator multiplicativo adicional de s^r para algum $s \in \mathbb{F}_{q^k}$, de modo que obtêm-se $e(P,Q) = a.n^s$ quando isso for calculado.

Da Propriedade A.1 (pág.79) tem-se que para qualquer $\xi \in \mathbb{F}_{q^k}$ existe $\xi^{q^k-1} = 1$, então se as^r for elevado a potência de $(q^k-1)/r$ obtêm-se

$$(a.s^r)^{(q^k-1)/r} = a^{(q^k-1)/r}.1 = a^{(q^k-1)/r}$$

de modo que uma exponenciação elimina o fator multiplicativo extra e deixa um resultado único. Assim, enquanto $\hat{e}(P,Q)$ não é único, a exponenciação adicional fornece

$$\hat{e}(P,Q)^{(q^k-1)/r}$$

que determina um valor único e, portanto, mais adequado para o uso. O uso de tais exponenciações para determinar um valor único é chamado de *exponenciação final* e o valor único é chamado de *emparelhamento reduzido*.

Na prática, muitas vezes precisa-se de um representante único para o valor do emparelhamento de Tate, e não uma classe de equivalência inteira, o que justifica a seguinte definição.

Definição 4.9 (Emparelhamento de Tate reduzido (M. Joye, 2008))**.** Seja E uma curva elíptica sobre $\mathbb{F}_q$ com $r|\#E(\mathbb{F}_q)$ e $mdc(r,q)=1$, e seja k o grau de mergulho, então o emparelhamento de Tate reduzido e_r é a aplicação

$$\begin{aligned} e_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) &\rightarrow \mu_r \subset \mathbb{F}_{q^k}^* \\ (P,Q) &\mapsto e_r(P,Q) = \hat{e}_r(P,Q)^{(q^k-1)/r}. \end{aligned}$$

Uma propriedade interessante de compatibilidade do emparelhamento de Tate reduzido é o seguinte: seja $r|N|(q^k-1)$, então para $P \in E(\mathbb{F}_{q^k})[r]$ e $Q \in E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ temos a igualdade $e_r(P,Q) = e_N(P,Q)$.

Finalmente, em muitos casos, $E(\mathbb{F}_{q^k})[r]$ e $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ são isomorfos.

De fato, desde que $E(\mathbb{F}_{q^k})[r] \bigcap rE(\mathbb{F}_{q^k}) = \{\mathcal{O}\}$, que é satisfeita se $E(\mathbb{F}_{q^k})$ não contém pontos de ordem r^2, temos a seguinte isomorfismo

$$\begin{aligned} \iota : E(\mathbb{F}_{q^k})[r] &\rightarrow E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}); \\ Q &\mapsto Q + rE(\mathbb{F}_{q^k}). \end{aligned}$$

Para acelerar o cálculo do emparelhamento de Tate, frequentemente restringe-se o primeiro argumento P para $E(\mathbb{F}_q)$.

O algoritmo de Miller corresponde a uma multiplicação escalar do ponto P, que é obviamente mais rápido se P for definido sobre $\mathbb{F}_q$, do que sobre algum corpo de extensão em $\mathbb{F}_{q^k}$.

4.2.1 Propriedades do emparelhamento de Tate

Para ver que o emparelhamento de Tate é linear em seu primeiro parâmetro, seja f_{P_1}, f_{P_2} e $f_{P_1+P_2}$ funções racionais de tal forma que temos

$$div(f_{P_1}) = r(P_1) - r(\mathcal{O}) div(f_{P_2}) = r(P_2) - r(\mathcal{O})$$

e

$$div(f_{P_1+P_2}) = r(P_1+P_2) - r(\mathcal{O})$$

Note que o divisor

$$D = (P_1+P_2) - (P_1) - (P_2) + (\mathcal{O})$$

é um divisor principal, assim é o divisor de alguma função racional, diz-se

$$div(g) = D$$

então

$$\begin{aligned} div(f_{P_1+P_2}) - div(f_1) - div(f_2) &= r(P_1 + P_2) - r(P_1) - r(P_2) - n(\mathcal{O}) \\ &= rD = rdiv(g) = div(g^r) \end{aligned}$$

de modo que

$$div(f_{P_1+P_2}) = div(f_1) + div(f_2) + div(g^r)$$

para que se possa escrever

$$f_{P_1+P_2} = f_1 f_2 g^r$$

Portanto

$$\begin{aligned} e(P_1 + P_2, Q) &= f_{P_1+P_2}(A_Q) = f_{P_1}(A_Q) f_{P_2}(A_Q) g^r(A_Q) \\ &= e(P_1, Q) e(P_2, Q) g^r(A_Q) \end{aligned}$$

Então, se as r-ésimas potências forem ignoradas, descobre-se que

$$e(P_1 + P_2, Q) = e(P_1, Q) e(P_2, Q)$$

como desejado.

Para ver que o emparelhamento de Tate é bilinear no segundo parâmetro, seja $A_{Q_1+Q_2}$ um divisor equivalente a $(Q_1 + Q_2) - (\mathcal{O})$, seja A_{Q_1} um divisor equivalente a $(Q_1) - (\mathcal{O})$ e seja A_{Q_2} um divisor equivalente a $(Q_1) - (\mathcal{O})$.

Então $A_{Q_1+Q_2} - A_{Q_1} - A_{Q_2}$ é equivalente a

$$D = (Q_1 + Q_2) - (Q_1) - (Q_2) + (\mathcal{O})$$

que é um divisor principal. Então $A_{Q_1+Q_2}$ é equivalente a $A_{Q_1} + A_{Q_2}$ porque eles diferem-se por um divisor principal. Assim, pode-se escrever

$$\begin{aligned} e(P, Q_1 + Q_2) &= f_P(A_{Q_1+Q_2}) \\ &= f_P(A_{Q_1} + A_{Q_2}) = f_P(A_{Q_1}) f_P(A_{Q_2}) \\ &= e(P, Q_1) e(P, Q_2) \end{aligned}$$

Uma aplicação que é não degenerada, bilinear e é também eficientemente computável é chamado de *emparelhamento*, e tais aplicações são as primitivas fundamentais a partir das quais muitos algoritmos criptográficos são construídos.

Por outro lado, o emparelhamento de Tate também tem a seguinte propriedade que limita a sua utilidade porque ele retorna o valor 1 em muitos casos.

Propriedade 4.2. *(Galbraith, 2001, sec.2.3 pág.500)* Seja $P \in E(\mathbb{F}_q)[r] \setminus \{O\}$ e r relativamente primo a q. Então, para ter $e(P, P) \neq 1$, devemos ter $k = 1$.

Assim, para um grau de mergulho $k > 1$ temos $e(P, P) = 1$, que também significa que

$e(aP, bP) = e(P, P)^{ab} = 1$ para inteiros a e b, de modo que o emparelhamento de Tate pode não parecer muito útil no início.

O seguinte resultado fornece uma visão sobre como superar essa limitação.

Propriedade 4.3. *(Verheul, 2004, sec.4.2 pág.204)* Seja n um número primo, $P \in E(\mathbb{F}_q)[r]$ $\{O\}$, para um $Q \in E(\mathbb{F}_{q^k})$ linearmente independente a P e $k > 1$. Então, tem-se que $e(P, Q)$ é não degenerada.

Então, se $P \in E(\mathbb{F}_q)[r]$ e um grau de mergulho não trivial, isto é, tem-se $k > 1$, então uma maneira de certificar-se que o emparelhamento de Tate $e(P, Q)$ é não degenerado é certificar-se que Q é linearmente independente a P.

Uma maneira de fazer isso é usar uma aplicação de distorção, de modo que em vez de computar $e(P, Q)$, calcula-se $e(P, \phi(Q))$, onde ϕ é uma aplicação de distorção apropriada.

Outra forma é calcular $e(P, \phi_d(Q))$, onde $Q \in E'$ está na torção da curva elíptica E e $\phi_d : E' \to E$ é a aplicação definida na Seção 3.3.1 (pág.18).

Em ambos os casos, denota-se o emparelhamento resultante por $\hat{e}(P, Q)$, onde $\hat{e}(P, Q) = e(P, \phi(Q))$ ou $\hat{e}(P, Q) = e(P, \phi_d(Q))$ conforme o caso, e denomina-se tais $\hat{e}$ de *emparelhamento de Tate modificado*.

4.3 Algoritmo de Miller

A técnica que usada na Seção 4.2 (pág.34) para encontrar um divisor equivalente a $r(P) - r(\mathcal{O})$, em que encontra iterativamente divisores equivalente a $(P) - (\mathcal{O})$, $2(P) - 2(\mathcal{O})$, ..., salvo $r(P) - r(\mathcal{O})$ por uma aplicação repetida da Propriedade 4.3, irá certamente funcionar, mas isso é extremamente ineficiente.

Em uma aplicação de criptografia, r é, tipicamente, pelo menos 2^{160}, de modo que a iteração dessa forma seja impraticável.

Em vez disso, o método para calcular $r(P) - r(\mathcal{O})$ é pela técnica de adição-e-duplicação[4], e encontrar um divisor equivalente a $r(P) - r(\mathcal{O})$. Este método é chamado de *algoritmo de Miller* (Miller, 2004).

O algoritmo de Miller é baseado na observação de que é fácil generalizar a Propriedade 4.3 para divisores.

$$D_1 = (aP) - (\mathcal{O}) + div(f_1)$$

e

$$D_2 = (bP) - (\mathcal{O}) + div(f_2)$$

ao descobrir que

$$D_1 + D_2 = (a + b)P - (\mathcal{O}) + div\left(f_1 f_2 \frac{u_{aP,bP}}{v_{(a+b)P}}\right)$$

Formaliza-se o algoritmo de Miller como se segue.

[4] tradução do inglês *double-and-add*

Escolha uma curva elíptica E em que todos os seguintes cálculos serão executados.
Seja $P \in E(\mathbb{F}_q)[r]$ e $Q \in E(\mathbb{F}_{q^k})$ com

$$r = \sum_{i=0}^{t} b_i 2^i s$$

de modo que $(b_i, \ldots, b_1, b_0)$ é a expansão binária de r.

Inicia-se com $f = 1$, $S = P$, e R um ponto aleatório em E. Então, faz-se uma iteração de adição-e-duplicação por meio da expansão binária de r, realizando a etapa de duplicação a cada iteração, e a etapa de adição se o *bit* atual é 1.

Isso permitirá construir a função racional equivalente a $r(P) - r(\mathcal{O})$ fora dos termos repetidamente dobrados, e valorar cada um destes termos em $(Q + R) - (R)$ como eles serão calculados.

Fazemos isso pelo seguinte algoritmo.

Algoritmo 1: Algorítmo de Miller para curvas elípticas

Entrada: Os pontos $P, R, T_1, T_2 \in E$, o índice r desejado
Saída: O valor $\frac{f_P(T_1)}{f_P(T_2)}$ onde $div f_P = r[P + R] - r[R] - [rP] + [\mathcal{O}]$

início
  Calcule $P + R$, a linha $\ell = ax + by + c$ por P e R,
  a linha vertical $v = x + d$ por $P + R$ e
  seja $g \leftarrow \dfrac{\left.\frac{ax+by+c}{x+d}\right|_{(x,y)=T_1}}{\left.\frac{ax+by+c}{x+d}\right|_{(x,y)=T_2}}$
  Seja $f \leftarrow g$, $J \leftarrow P$, $j \leftarrow 1$
  Escreva $r = (r_{n-1}, \ldots, r_1, r_0)$ em base binária
  para $i = n - 2$; $i \geq 0$; $i - -$ **faça**
    Seja $\ell = ax + by + c$ a tangente em J
    $S \leftarrow 2J$
    Seja $v = x + d$ a linha vertical que passa por S
    Seja $f \leftarrow f^2 \cdot \left.\frac{\ell}{v}\right|_{T_1} \cdot \left.\frac{\ell}{v}\right|_{T_2}$
    $J \leftarrow S, j \leftarrow 2j$
    se $r_i = 1$ **então**
      Seja $\ell = ax + by + c$ a linha que passa por J e P
      $S \leftarrow J + P$
      Seja $v = x + d$ a linha vertical que passa por S
      Seja $f \leftarrow f \cdot g \cdot \left.\frac{\ell}{v}\right|_{T_1} \cdot \left.\frac{\ell}{v}\right|_{T_2}$
      $J \leftarrow S, j \leftarrow j + 1$
  devolve f
fim

A idéia do algoritmo de Miller é poder encontrar f_{j+k} através de f_j e f_k, onde f_i representa a função racional cujo divisor correspondente é:

$$D_i = i[P+R] - i[R] - [iR] + [\infty]$$

(Martin, 2008, sec.4.3 pág.84) (M. Joye, 2008).

Tal divisor foi criado porque $D_r = r[P+R] - r[R] - [rR] + [\infty]$, e como $P \in E[r]$, $rP = P$ e portanto $D_r = r[P+R] - r[R]$, como deseja-se. Além disso, cada D_i é um divisor principal, de maneira que existe uma função f_i correspondente. Agora pode-se observar como é possível construir essas funções de modo a obter $f_r(T_1)/f_r(T_2)$ (Washington, 2008, sec.11.4 pág.361).

Supondo que $f_j(T_1)/f_j(T_2)$ e $f_k(T_1)/f_k(T_2)$ já foram calculados. Seja $ax+by+c=0$ a equação da reta que passa por jP e kP e seja $x+d=0$ a reta vertical que passa por $(j+k)P$, então

$$div\left(\frac{ax+by+c}{x+d}\right) = [jP] + [kP] - [(j+k)P] - [\infty]$$

Logo,

$$D_{j+k} = D_j + D_k + div\left(\frac{ax+by+c}{x+d}\right)$$

Portanto,

$$f_{j+k} = \gamma f_j f_k \frac{ax+by+c}{x+d}$$

Assim,

$$\frac{f_{j+k}(T_1)}{f_{j+k}(T_2)} = \frac{f_j(T_1)}{f_j(T_2)} \cdot \frac{f_k(T_1)}{f_k(T_2)} \cdot \frac{\left.\frac{ax+by+c}{x+d}\right|_{(x,y)} = T_1}{\left.\frac{ax+by+c}{x+d}\right|_{(x,y)} = T_2}$$

A propriedade do emparelhamento utilizando as técnicas apresentadas anteriormente (Associatividade).

$$\begin{aligned} e(P_1+P_2,Q) &= e(P_1,Q).e(P_2,Q) \text{ e} \\ e(P,Q_1+Q_2) &= e(P,Q_1).e(P,Q_2) \end{aligned}$$

Exemplo 4.2. Utilizando a curva $E(\mathbb{F}_{43}) : y^2 = x^3 + x$ e os parâmetros $p = 43$, $q = 11$ e $l = 4$. E os pontos $P_1 = (23, 8)$, $P_2 = (31, 18)$ e $Q = (14, 36)$.

$$\begin{aligned} e(P_1+P_2,Q) &= e(P_1,Q).e(P_2,Q) \\ e((4,5),(14,36)) &= e((23,8),(14,36)).e((31,18),(14,36)) \\ 7+9i &= (26+23i).(2+30i) \\ 7+9i &= 7+9i \end{aligned}$$

E a segunda propriedade:

$$\begin{aligned} e(aP,Q) &= e(P,Q)^a = e(P,aQ) \\ e(xP+yP,Q) &= e(xP,Q).e(yP,Q). \end{aligned}$$

onde a, x e $y \in \mathbb{Z}$

Exemplo 4.3. Utilizando a mesma curva do exemplo anterior, com o parâmetro escalar $a = 761$ e os pontos $P = (23, 35)$ e $Q = (14, 36)$.

$$\begin{aligned} e(aP, Q) &= e(P, aQ) \\ e(761(23, 35), (14, 36)) &= e((23, 35), 761(14, 36)) \\ e((14, 7), (14, 36)) &= e((23, 35), (31, 25)) \\ 18 + 8i &= 18 + 8i \end{aligned}$$

e

$$\begin{aligned} e(aP, Q) &= e(P, Q)^a \\ e(761(23, 35), (14, 36)) &= e((23, 35), (14, 36))^{761} \\ e((14, 7), (14, 36)) &= e((23, 35), (14, 36))^{761} \\ 18 + 8i &= 18 + 8i \end{aligned}$$

4.4 Emparelhamento de Ate

O emparelhamento Ate (Hess *et al.*, 2006), juntamente com emparelhamento η (Eta) (Barreto *et al.*, 2007), são os mais eficientes métodos para calcular o emparelhamento de Tate reduzido, quando restrito a autoespaços de Frobenius.

Lema 4.2. Seja E uma curva elíptica sobre $\mathbb{F}_q$ e seja $r|\#E(\mathbb{F}_q)$, com $mdc(r, q) = 1$ e grau de mergulho k. Seja $\lambda = Cr$ um múltiplo de r, então a seguinte aplicação

$$\begin{aligned} a_\lambda : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) &\rightarrow \mu_r \subset \mathbb{F}_{q^k}^* : \\ (P, Q) &\mapsto a_\lambda(P, Q) = f_{\lambda,P}(Q)^{(q^k-1)/r}, \end{aligned}$$

com $f_{\lambda,P}$ normalizado, define um emparelhamento bilinear que é não-degenerado se e somente se $mdc(r, C) = 1$.

O lema acima pode ser modificado da seguinte forma: seja $N = mdc(q^k - 1, \lambda)$ e $C = \lambda/N$, então $f_{\lambda,Q}^{(q^k-1)/N}$ define um emparelhamento bilinear que é não-degenerada se e somente se $mdc(r, C) = 1$.

A vantagem desta formulação é que, para escolhas particulares de N, a exponenciação final $(q^k - 1)/N$ tem peso Hamming baixo na base q, que pode ser usado para acelerar a computação.

Os emparelhamentos Ate e Eta (η) são então obtidos simplesmente escolhendo um λ especial e explorando a ação de um endomorfismo eficientemente computável, como Frobenius, em $\mathbb{G}_1$ e $\mathbb{G}_2$.

O emparelhamento Eta é definido sobre curvas supersingulares, e elas possuem baixo grau de mergulho com $k \leq 6$ para curvas elípticas. Por esse motivo o estudo delas fogem do escopo deste trabalho.

Todos os emparelhamentos Ate em curvas elípticas ordinárias são derivados do emparelhamento de Tate, restringido a $\mathbb{G}_2 \times \mathbb{G}_1$ e explorando $\pi_q(Q) = [q]Q$ para $Q \in \mathbb{G}_2$ e $\pi_q(P) = P$ para $P \in \mathbb{G}_1$.

Pela definição do grau de mergulho temos $r|(q^k - 1)$ e, portanto, $r|(S^k - 1)$ para qualquer inteiro $S \equiv q \pmod r$. Isso leva ao seguinte teorema.

Teorema 4.1 (Emparelhamento Ate)**.** *(M. Joye, 2008, versão I pág.25)* Seja S qualquer inteiro com $S \equiv q \pmod r$. Seja $\lambda = S^k - 1$ e $C = \lambda/r$, então

$$\begin{aligned} a_S : \mathbb{G}_2 \times \mathbb{G}_1 &\rightarrow \mu_r \subset \mathbb{F}_{q^k}^* : \\ (Q, P) &\mapsto a_S(Q, P) = f_{S,Q}(P)^{(q^k-1)/r}, \end{aligned}$$

com $f_{S,Q}$ normalizado define um emparelhamento bilinear que é não-degenerado se e somente se $mdc(r, C) = 1$.

Da mesma forma para Lema 4.2, o Teorema 4.1 pode ser adaptado usando uma exponenciação final diferente da seguinte forma: seja $N = mdc(q^k - 1, \lambda)$ e $C = \lambda/N$, então $f_{S,Q}^{c_S(q^k-1)/N}$ com $c_S = mdc(N, \sum_{i=0}^{k-1} S^{k-1-i}q^i)$ define um emparelhamento bilinear que é não-degenerado se e somente se $mdc(r, C) = 1$.

O Teorema 4.1 pode ser estendido imediatamente observando que $r|((q^i)^k - 1)$ para todos os inteiros positivos i, então a condição de S no teorema pode ser relaxado para $S \equiv q^i \pmod r$.

A principal vantagem do emparelhamento Ate sobre o emparelhamento de Tate é que S pode ser muito menor do que r e, portanto, leva a um ciclo muito mais curto no algoritmo de Miller.

No entanto, este ganho é compensado totalmente pelo fato de que Q é definido sobre $\mathbb{F}_{q^k}$ e não sobre $\mathbb{F}_q$.

Felizmente, é possível representar $\mathbb{G}_2$ por um subgrupo de torção como na Propriedade 3.2 (pág.19), que é definido sobre uma pequena extensão de $\mathbb{F}_q$.

O efeito líquido é que o emparelhamento Ate para o pequeno S e usando torções será mais rápido do que o emparelhamento de Tate correspondente.

Isso, automaticamente leva à questão do quão pequeno, S pode ser.

Note que

$$r|\Phi_{k/d}(q^i), \text{ onde } d = mdc(i, k),$$

o que implica que o valor mínimo para $q^i \bmod r$ é $r^{1/\varphi(k/d)}$.

Lembre-se que Φ_k denota o k-ésimo polinômio ciclotômico.

Para $mdc(i, k) = 1$ que, portanto, obtém o menor limite inferior de aproximadamente $r^{1/\varphi(k)}$.

Este limite ideal é atingido para algumas famílias de curvas de emparelhamento amigável, mas não em geral.

O Teorema 4.1 é uma aplicação bastante limitada do Lema 4.2 em que só são considerados múltiplos de r da forma $S^k - 1$ para um S bem escolhido.

Ao considerar todos os possíveis múltiplos de r e explorando o fato de que o produto e também a divisão de dois emparelhamentos (desde que o emparelhamento tenha ordem r) é novamente um emparelhamento, obtemos a seguinte construção (Hess, 2009; Vercauteren, 2008, sec.B pág.2).

Para $h = \sum_{i=0}^{d} h_i z^i \in \mathbb{Z}[z]$ com $h(S) \equiv 0 \bmod r$, sendo que $f_{S,h,Q} \in \mathbb{F}_{q^k}(E)$ denota a função normalizada com divisor

$$div(f_{S,h,Q}) = \sum_{i=0}^{d} h_i((S^iQ) - (\mathcal{O}))$$

e seja $||h||_1 = \sum_{i=0}^{d} |h_i|$.

O seguinte teorema que generaliza o Teorema 4.1 que foi provado em Hess (2009) (teo. 1).

Teorema 4.2 (Emparelhamento Ate). *(M. Joye, 2008, versão II pág.26)* Suponha que S é uma k-ésima raiz da unidade primitiva modulo r^2, então

$$\begin{aligned} a_{S,h} : \mathbb{G}_2 \times \mathbb{G}_1 &\to \mu_r \subset \mathbb{F}_{q^k}^*; \\ (Q, P) &\mapsto (f_{S,h,Q}(P))^{(q^k-1)/r} \end{aligned}$$

define um emparelhamento bilinear que é não-degenerado, se e somente se $h(S) \equiv 0 \bmod r^2$.

A relação com o emparelhamento de Tate reduzido é $a_{S,h}(Q,P) = e_r(Q,P)^{h(S)/r}$.

Qualquer $h \in \mathbb{Z}[z]$ de tal forma que o emparelhamento acima é não-degenerado satisfaz $||h||_1 \geq r^{1/\varphi(k)}$.

Para usar o Teorema 4.2 na prática, precisamos encontrar um h com $||h||_1$ pequeno, e maior que $r^{1/\varphi(k)}$, senão o emparelhamento é degenerado.

Como um polinômio pode ser encontrado por vetores curtos no seguinte reticulado de dimensão m (com $\varphi(k) \leq m \leq n$) (gerado pelas linhas)

$$L := \begin{pmatrix} r & 0 & 0 & \dots & 0 \\ -S & 1 & 0 & \dots & 0 \\ -S2 & 0 & 1 & \dots & 0 \\ \dots & \dots & & \ddots & \\ -S^{m-1} & 0 & \dots & 0 & 1 \end{pmatrix}.$$

Algum vetor curto $(w_0, w_1, \dots, w_{m-1})$ tal que $h = \sum_{i=0}^{m-1} w_i z^i$ satisfaz $||h||_1 \geq r^{1/\varphi(k)}$ e $h(S) \not\equiv 0 \bmod r^2$ dá origem a um emparelhamento não-degenerado.

Para famílias de curvas de emparelhamento amigável (ver Seção 4.5), r e q são obtidos como a valoração de polinômios $r(x)$ e $q(x)$.

Como tal, os vetores curtos em L também podem ser parametrizados por polinômios em x, que mostra que os coeficientes h_i de h no Teorema 4.2 estão relacionados e isso proporciona uma maior simplificação do cálculo de emparelhamento.

Os Teoremas 4.1 e 4.2 definem o emparelhamento Ate em $\mathbb{G}_2 \times \mathbb{G}_1$ e, como observado antes, isso realmente necessita de torções para acelerar os cálculos no $\mathbb{G}_2$.

Esta complicação poderia ser evitada completamente se uma construção semelhante fosse possível em $\mathbb{G}_1 \times \mathbb{G}_2$.

No entanto, para curvas elípticas ordinárias, isso só é possível em um sentido restrito.

O teorema a seguir deve ser comparado com o Teorema 4.1. A prova é exatamente a mesma que a prova do Teorema 4.1 substituindo π_q por um endomorfismo diferente puramente inseparável $\psi \circ \pi_q^e$, com $\psi \circ \pi_q^e(Q) = Q$ para $Q \in \mathbb{G}_2$ e $\psi \circ \pi_q(P) = [q]P$ para $P \in \mathbb{G}_1$.

Exemplo 4.4. Utilizando a curva BN $E(\mathbb{F}_p) : y^2 = x^3 + 3$, e a curva BN torcida $E'(\mathbb{F}_p) : y'^2 = x'^3 + (3,3)$ de 56 bits, com o parâmetro $p = 5755$ e os pontos $P = (32479380864452404, 2789781853982508, 993663864399096)$ e $Q = (28243273031411577 + 21432891735600992i,$
$8187581076509695 + 32721277577247448i,$
$27649639615870957 + 3770041453320208i)$.

$$\begin{aligned} a_{S,h}(P,Q) &= (f_{S,h,Q}(P))^{(q^k-1)/r} \\ &= ((18616148340923551, 29755583311034270), (22624598026079760, 4154096988649059), \\ &\quad (30889204818197104, 16781347605245021), (17404110572989392, 31790030444992789), \\ &\quad (22300664094602286, 30881136583138645), (29674150004718990, 26653662516545883)) \end{aligned}$$

Executado em 42,6 ms.

Teorema 4.3 (Emparelhamento Ate torcido)**.** *(M. Joye, 2008, pág.27)* Assume-se que E admite uma torção de grau d e seja $z = mdc(d,k)$ e $e = k/z$.

Seja S qualquer inteiro com $S \equiv q^e \pmod{r}$ e defina $\lambda = S^z - 1$ e $C = \lambda/r$, então

$$\begin{aligned} a_S^t : \mathbb{G}_1 \times \mathbb{G}_2 &\rightarrow \mu_r \subset \mathbb{F}_{q^k}^*; \\ (P, Q) &\mapsto a_S^t(P, Q) = f_{S,P}(Q)^{(q^k-1)/r}, \end{aligned}$$

define um emparelhamento bilinear que é não-degenerado, se e somente se $mdc(r, C) = 1$.

O Teorema 4.2 pode ser adaptado, substituindo todos os q's por q^e's.

A principal diferença com o Teorema 4.1 é que $S \equiv q^e \pmod{r}$, com $e = k/z$.

O parâmetro S pode ser tão pequeno como $r^{\varphi(z)}$ e na maioria dos casos, temos $z < k$ (M. Joye, 2008).

Para $k = d$, obtêm-se um emparelhamento melhor e mais rápido do que o emparelhamento Ate convencional.

Exemplo 4.5. Utilizando a curva BN $E'(\mathbb{F}_p) : y'^2 = x'^3 + (3, 3)$ de 56 bits, com o parâmetro $p = 5755$ e os pontos $P = (32479380864452404, 2789781853982508, 993663864399096)$ e
$Q = (28243273031411577 + 21432891735600992i,$
$8187581076509695 + 32721277577247448i,$
$27649639615870957 + 3770041453320208i)$.

$$\begin{aligned} a_S^t(P, Q) &= f_{S,P}(Q) \\ &= ((9680625868253750, 37562019742392643), (31555778860340199, 36861108485177386), \\ &\quad (30036865503298157, 13886713046734785), (36164473734001995, 6306784855184279), \\ &\quad (5356995592834535, 7528766313060466), (8526100821669007, 3609060276273025)) \end{aligned}$$

Executado em $44,57$ ms.

4.5 Curvas elípticas ordinárias de emparelhamento amigável

Esta seção dá uma breve visão geral dos algoritmos mais importantes para construir curvas elípticas ordinárias de emparelhamento amigável.

A necessidade de construções especializadas já foi apresentada na Seção 4.4 (pág.41): para uma curva elíptica E sobre $\mathbb{F}_q$ e r um divisor primo de $\#E(\mathbb{F}_q)$, com $mdc(r, q) = 1$, o grau de mergulho k foi definido como a ordem de q em $(\mathbb{Z}/r\mathbb{Z})^*$.

Assim, em geral, k irá ser tão grande quanto r o que torna a computação no corpo finito $\mathbb{F}_{q^k}$ impossível.

Além disso, q, r e k devem ser escolhidos de tal forma que o esforço tomado pelos melhores ataques contra o problema do logaritmo discreto em um subgrupo de ordem r de $E(\mathbb{F}_q)$, por exemplo, usando ρ (rho) de Pollard, é equilibrado com o esforço tomado pelo melhor ataque ao problema do logaritmo discreto em $\mathbb{F}_{q^k}^*$ (Pollard, 1978).

Por exemplo, para atingir um nível de segurança de 128 bits deve-se escolher $r \simeq 2^{256}$ e $q^k \simeq 2^{3072}$.

O método MC, apresentado na Seção 3.7.1 (pág.26), constrói uma curva elíptica E sobre $\mathbb{F}_q$ com um determinado número de pontos $\#E(\mathbb{F}_q) = q + 1 - t$. Ao tomar o discriminante da equação característica de Frobenius, obtêm-se a equação norma de MC

$$4q - t^2 = \delta V^2,$$

onde δ é chamado de discriminante e é inteiro positivo e sem fator quadrático[5] para t ímpar ou da forma $4d$ com d livre de quadrados para t par.

Dado q e t, o método MC será eficiente para construir uma curva elíptica E sobre $\mathbb{F}_q$ com $q+1-t$ pontos, somente quando δ for suficientemente pequeno, por exemplo, $\delta < 10^{10}$.

A proposição a seguir fornece uma caracterização simples do grau de mergulho em termos de traço de Frobenius e está no cerne de todas as construções.

Propriedade 4.4. Seja E uma curva elíptica sobre $\mathbb{F}_q$ com $\#E(\mathbb{F}_q) = q+1-t = hr$, com r primo.

Seja k um inteiro positivo com $r \not| k$, então k é o grau de mergulho com respeito a r se e somente se $\Phi_k(q) \equiv 0 \pmod{r}$ ou equivalentemente $\Phi_k(t-1) \equiv 0 \pmod{r}$.

Um método muito poderoso para a construção de curvas elípticas de emparelhamento amigável é a parametrização de t, r, q por polinômios $t(x)$, $r(x)$, $q(x)$ e exigir que estes satisfaçam às propriedades correspondentes: $r(x)|q(x)+1-t(x)$, $r(x)|\Phi_k(t(x)-1)$ e $\delta V^2 = 4q(x) - t(x)^2$ tenha infinitas soluções inteiras.

4.6 Curvas BN (Barreto-Naehrig)

Uma curva BN é uma curva elíptica $E_b : y^2 = x^3 + b$ definida sobre $\mathbb{F}_q$ tal que o grupo de pontos $\mathbb{F}_q$-racional tem ordem prima r. As curvas BN tem grau de mergulho $k = 12$ em relação ao r (Barreto *et al.*, 2005).

A família de curvas BN de emparelhamento amigável é parametrizada pelos seguintes polinômios na variável x:

$$p = p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1,$$
$$r = r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1.$$

O vetor mais curto no reticulado L para a norma Euclidiana é

$$V_1(x) = [x+1, x, x, -2x],$$
$$V_2(x) = [2x, x+1, -x, x].$$

Desde que exista uma relação entre $h_i(x)$, o cálculo da função $f_{S,h,Q}$ do Teorema 4.2 (pág.42) segue de $f_{x,Q}$. Por outro lado, pode-se interpretar esses vetores curtos com o número mínimo de coeficientes de tamanho x e obter

$$W(x) = [6x+2, 1, -1, 1]$$

Desde que $f_{1,Q} = 1$ e $f_{-1,Q} = 1/f_{1,QvQ}$ (que se anula depois da exponenciação final), o emparelhamento $a_{S,h}$ é calculado por:

$$a_{S,h} = (f_{6x+2,Q}(P) \cdot l_{Q_3,-Q_2}(P) \cdot l_{-Q_2+Q_3,Q_1}(P) \cdot l_{Q_1-Q_2+Q_3,[6x+2]Q})^{(q^k-1)/r},$$

[5] Um inteiro *sem fator quadrático* ou *livre de quadrados* ou, ainda, um *quadratfrei*, é um número inteiro que não é múltiplo de nenhum quadrado perfeito.

onde $Q_i = Q^{q^i}$ para $i = 1, 2, 3$.

Barreto *et al.* (2005) notaram que se $r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1$ é um dos fatores, o traço de Frobenius sobre $\mathbb{F}_p$ é $t(x) = 6x^2 + 1$ e portanto $q(x) = r(x) + t(x) - 1$, então a equação norma de MC é

$$t(x)^2 - 4q(x) = -3(1 + 4x + 6x^2)^2,$$

que mostra que as curvas BN tem discriminante $\delta = 3$.

Para encontrar uma curva BN basta escolher valores inteiros do tamanho adequado para x até que $p(x)$ e $r(x)$ sejam primos.

Em seguida, testar valores para o coeficiente b até que a curva tenha a ordem correta.

Os detalhes da implementação serão discutidos no Capítulo 6.

4.7 Resumo

Neste capítulo foram apresentados os principais tipos de emparelhamentos utilizados.

Dentre eles, emparelhamento de Weil, Tate e Ate, e apresentamos o algoritmo de Miller utilizado para calcular emparelhamentos.

Capítulo 5

Protocolos implementados

Neste capítulo, vamos apresentar os protocolos de acordo de chaves estudados, que serviram de modelo e as adaptações realizadas para utilização no Projeto Borboleta.

Os protocolos de acordo de chaves com autenticação no modelo de criptografia de chave pública sem certificados (CL-AKA) permitem que dispositivos de baixa capacidade computacional e um servidor se autentiquem mutuamente e estabeleçam chaves de sessão não falsificáveis. Essas chaves de sessão podem ser adotadas como chaves secretas em cifras de bloco, para garantir sigilo nas comunicações.

Nesse contexto, foi adotado um protocolo de acordo de chaves para o mecanismo de transferência segura dos dados dos prontuários médicos.

A evolução dos protocolos criptográficos forneceu uma estrutura de evolução adaptativa para os esquemas.

Dessa forma, um modelo de chave pública convencional era modificado para atender os requisitos de um modelo baseado em identidade, este por sua vez era modificado para um modelo sem certificados.

Muitos trabalhos foram feitos neste sentido. Por exemplo, para a encriptação sem certificado, foi combinado um modelo baseado em chave pública com um modelo baseado em identidade, como visto em Libert e jacques Quisquater (2006).

De mesmo modo, as assinaturas sem certificados e mecanismos de encapsulamento de chave sem certificados de Bentahar *et al.* (2008) podem ser construídos a partir de protocolos existentes.

Para os protocolos de acordos de chaves, ao contrário do esperado, os autores Lippold *et al.* (2009) mostraram que um protocolo de acordo de chave sem certificado não pode ser construído de forma segura por uma combinação natural de um protocolo de acordo de chave baseado em identidade com um protocolo de acordo de chave baseado no modelo fundamental de chave pública.

O modelo de segurança proposto por Lippold *et al.* (2009) para acordo de chave sem certificado é uma extensão de Swanson (2008), versão modificada do modelo estendido de Canetti e Krawczyk, apresentado em LaMacchia *et al.* (2007).

Este modelo é ainda mais forte que o de Swanson, dando mais poder ao adversário.

Os autores fornecem a primeira prova formal de um esquema fortemente seguro de acordo de chaves sem certificado, demonstrado no modelo do oráculo aleatório.

Foi demonstrado por Lippold *et al.* (2009) que este protocolo de acordo de chaves sem certificados é seguro mesmo se o Centro de Geração de Chaves (KGC) tentar quebrar ativamente o esquema: ele pode tanto revelar os segredos efêmeros ou revelar valores secretos e substituir chaves

públicas, mas não ambos.

Na verdade, desde que cada participante possua pelo menos um segredo não comprometido, o sistema ainda será seguro no modelo do oráculo aleatório presumido que a hipótese Diffie-Hellman computacional e a hipótese bilinear Diffie-Hellman computacional sejam mantidas.

As provas estão nos modelos de segurança mais fortes disponíveis para esquemas sem certificados, ou seja, corresponde a segurança forte de Dent (2008) (sec.2 pág.2) do Tipo I e Tipo II, onde o adversário tem a permissão de substituir as chaves públicas sem certificado e o desafiante ainda tem que responder a todas as consultas do oráculo.

Além disso, o protocolo proposto por Lippold *et al.* (2009) é um protocolo de uma rodada[1] que resiste a todos os ataques propostos de Swanson.

Embora as mensagens trocadas no protocolo sejam exatamente as mesmas mensagens do protocolo de Mandt e Tan (2006), para suportar os ataques que usam uma versão modificada da técnica apresentada por Xia *et al.* (2008).

5.1 Modelo de segurança para esquemas de acordo de chave sem certificados

As propriedades de segurança mais importantes e requeridas nos protocolos de acordo de chaves com autenticação são relacionadas a seguir. Os protocolos aqui mencionados contemplam todas elas.

Resistência a ataques de personificação básicos : um adversário não deve ser capaz de personificar um usuário se não conhecer sua chave secreta.

Resistência a ataques de compartilhamento desconhecido de chave (UKS *Unknown Key-Share*): deve ser inviável convencer um participante A de que ele está compartilhando uma chave com B, quando na realidade está compartilhando com outro usuário C (honestamente registrado no sistema), enquanto C pensa (corretamente) estar compartilhando com A.

Segurança de chave conhecida : cada execução do protocolo deve produzir uma chave de sessão única. O protocolo deve permanecer seguro mesmo que um adversário descubra algumas chaves de sessão já negociadas (Law *et al.*, 2003).

Segurança no futuro completa-fraca (wPFS, ou *Weak Perfect Forward Secrecy*) : deve ser inviável para um atacante recuperar uma chave de sessão mesmo que, no futuro, venha a corromper as chaves secretas de longa duração de *todos* os participantes envolvidos naquela sessão, porém sob a condição de que ele não esteve ativamente envolvido na escolha dos segredos temporários para o cálculo daquela chave de sessão. Esta última condição é o que caracteriza a segurança PFS como *fraca*. Em Krawczyk (2005) (def.22) foi demonstrado que em protocolos com apenas duas passagens de mensagens (como é o caso de todos os aqui discutidos) wPFS é o melhor que se pode alcançar com relação à segurança no futuro.

Resistência a ataques de personificação pelo comprometimento de chave secreta (KCI, ou *Key-Compromise Impersonation*): se a chave secreta de longa duração de um usuário

[1] Protocolo não iterativo

A é comprometida, um atacante não deve ser capaz de personificar um usuário B perante A (Krawczyk, 2005, def. 20).

Resistência ao vazamento de segredos temporários : o vazamento de um valor secreto temporário não deve comprometer a segurança de sessões que não o tenham usado.

No caso especial em que não são necessários certificados digitais para as chaves públicas, duas propriedades adicionais são desejáveis:

Segurança no futuro perante o KGC (*KGC Forward Secrecy*) : o KGC deve ser incapaz de recuperar chaves de sessão, mesmo que monitore o tráfego durante o estabelecimento das chaves e que, portanto, tenha acesso a todos os dados públicos.

Para protocolos sem certificados, Mandt e Tan (2006) exige uma propriedade adicional chamada de "*resistência a informações temporárias conhecidas de sessão específica*", mas ele fornece apenas uma definição informal.

Se o KGC conhece os segredos temporários com todas as entradas para a chave de sessão, não é possível fornecer essa propriedade em um esquema de acordo de chaves baseado em identidade.

Resistência ao vazamento de segredos temporários para o KGC : nos protocolos CL-AKA, o KGC ainda que seja capaz de aprender os valores secretos temporários de qualquer sessão (e fazendo uso de seu conhecimento de todas as chaves secretas parciais) deve ser incapaz de recuperar a chave de sessão.

Os protocolos estudados foram demonstrados seguros segundo o modelo de Lippold *et al.* (2009), que é uma extensão do modelo CK, conhecida por eCK e definida em LaMacchia *et al.* (2007) (def.3.2 pág.7).

Swanson (2008) e Lippold et al. estenderam eCK para incluir adaptações para o caso sem certificado.

Os modelos CK e eCK diferem fundamentalmente no modo como são tratados os vazamentos dos segredos temporários. Enquanto em CK *todos* os segredos temporários de uma sessão podem ser revelados de uma única vez (por meio do oráculo *SessionStateReveal*), em eCK os temporários podem ser comprometidos um a um, à escolha do adversário.

Como já foi mencionado, a demonstração de segurança, não é o assunto principal deste trabalho, mas a sua existência é uma parte importante para a concepção do mesmo.

5.2 Arquitetura dos Protocolos

Definição 5.1. No modelo de segurança, uma sessão é considerada "*recente*"[2], se cada participante ainda possua pelo menos um segredo não comprometido.

Uma composição de um esquema de Estabelecimento de chaves autenticado baseado em identidade (ID-AKE) com um esquema de Estabelecimento de chaves autenticado baseado em chave pública tradicional (PK-AKE) está representado na Figura 5.1.

As linhas indicam que combinação de segredos oferece resistência contra qual tipo de ataque.

[2]tradução do inglês "*fresh session*"

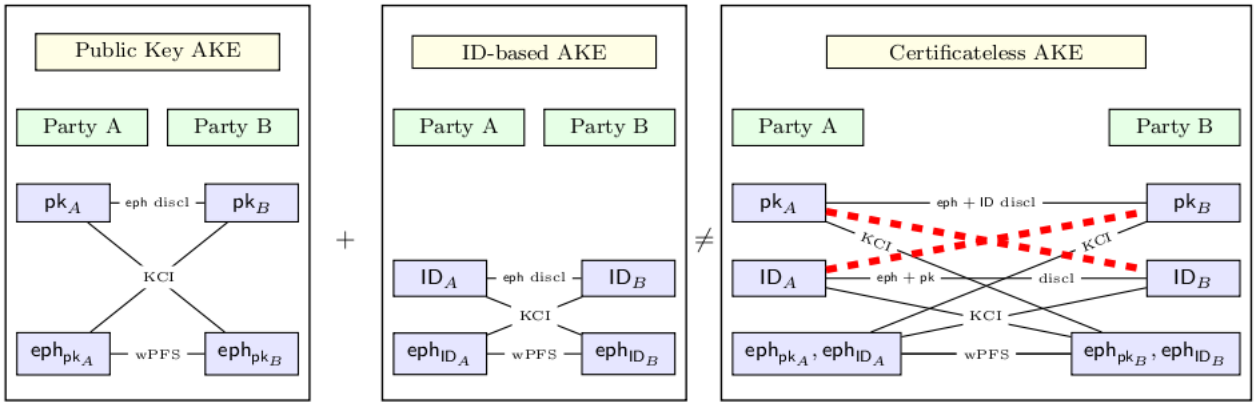


Figura 5.1: *PK-AKE + ID-AKE ≠ CL-AKE*

Um caminho natural para alcançar tal composição consiste em executar os dois protocolos em paralelo e derivar a chave de sessão da composição global, em função do conhecimento de somente duas componentes da chave de sessão.

Esta composição não pode oferecer o nível de segurança desejado, porque não existem garantias de segurança, se o participante A ainda possua uma chave não comprometida no PK-AKE, e se o participante B ainda possua uma chave não comprometida no ID-AKE.

Neste momento ambos esquemas AKE estão quebrados, pois este modelo não oferece resistência ao vazamento de segredos temporários para o KGC nas linhas pontilhadas na Figura 5.1.

Isto pode explicar porque os esquemas de estabelecimento de chaves autenticado sem certificados (CL-AKE) com um prova de segurança não foram publicados antes de Lippold *et al.* (2009).

Exemplo para sistemas de chaves públicas aplicáveis a este diagrama seria NAXOS (LaMacchia *et al.*, 2007) e CMQV (Ustaoglu, 2008), um exemplo para um sistema baseado em identidade seria o esquema ASIACCS09 (Huang e Cao, 2009a).

5.3 Descrição do protocolo de Lippold-Boyd-González

Nesta seção, é descritas as fases do protocolo de Lippold *et al.* (2009) de troca de chaves sem certificado autenticado.

O protocolo consiste em três fases: instalação; troca de mensagens ou geração de chaves do usuário; e a computação da chave de sessão ou acordo de chave.

5.3.1 Instalação

O KGC publica um gerador $P \in \mathbb{G}$ e um emparelhamento bilinear admissível $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ que preenche os critérios definidos na seção anterior.

Os grupos algébricos de emparelhamento adequado para este protocolo devem ser emparelhamentos do Tipo 1 e 4 descritos na Seção 4 (pág.29) (Cheng *et al.*, 2007).

Neste protocolo, acredita-se que os emparelhamentos assimétricos não são possíveis porque Lippold *et al.* (2009) utiliza o acordo de chave não interativo baseado em identidade de Sakai *et al.* (2000) como parte do protocolo.

Como o protocolo de Sakai *et al.* (2000) utiliza-se de propriedades específicas de emparelhamentos simétricos.

Isto é, ele requer uma função de hash para ambos $\mathbb{G}$ e $\mathbb{G}_T$.

O protocolo de SOK foi provado seguro por Dupont e Enge (2002) usando a hipótese de "*Gap Diffie-Hellman*".

O KGC escolhe um valor aleatório $s \in \mathbb{Z}_p$ como a chave mestra secreta e calcula sP como a sua chave pública.

O KGC seleciona três funções de hash criptográficas.

$$\begin{aligned} H: &\quad \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}^8 \times \mathbb{G}_T^6 \to \{0,1\}^n \text{ para algum inteiro } n > 0 \\ H_1: &\quad \{0,1\}^* \to \mathbb{G} \\ H_2: &\quad \mathbb{G} \to \mathbb{G} \end{aligned}$$

onde H é a função chave de derivação do esquema.

Cada usuário U gera um valor secreto $x_U \xleftarrow{\$} \mathbb{Z}_p$ e uma chave pública $x_U P \in \mathbb{G}$.

Cada usuário U informa sua chave secreta baseada na identidade $\{H_1(ID_U), H_2(H_1(ID_U))\} \in \mathbb{G}^2$ para o centro gerador de chave (KGC). E o KGC devolve $\{sH_1(ID_U), sH_2(H_1(ID_U))\} \in \mathbb{G}^2$ com a chave mestre s implícita.

5.3.2 Troca de mensagens

Para estabelecer uma chave compartilhada, usuário A gera o segredo efêmero $x_A \xleftarrow{\$} \mathbb{Z}$ e o usuário B gera o segredo efêmero $x_B \xleftarrow{\$} \mathbb{Z}$. Eles trocam as mensagens em seguida:

$$\begin{aligned} A \to B: &\quad E_A = (r_A P, x_A P) \\ B \to A: &\quad E_B = (r_B P, x_B P) \end{aligned}$$

Nota-se que as chaves públicas sem certificados podem ser obtidas das mensagens se forem publicadas em um diretório público online.

Isto poupará largura de banda, mas ao mesmo tempo pode tornar o sistema mais vulnerável ao equivalente *ataque da negação de decifração*[3] na criptografia sem certificado, pois um adversário pode manipular as entradas do diretório mais facilmente do que a troca da mensagem entre dois participantes.

Como o protocolo de Lippold *et al.* (2009) é de rodada única, ele consegue somente a autenticação implícita. Krawczyk (2005) (sec.2 pág.4) mostra que a autenticação explícita é possível com três rodadas e meia.

A seguir, implicitamente, exige-se que cada um dos participantes sempre verifique o conjunto dos membros do subgrupo para todos os elementos das mensagens que são trocadas no protocolo para se defender contra o ataque de subgrupos pequenos.

5.3.3 Cálculo das chaves

Para computar a chave de sessão, o participante A calcula:

Cálculo de K_A e K'_A:

[3] Trata-se de um ataque em que o adversário substitui chaves públicas e induz os remetentes a cifrarem mensagens com falsas chaves públicas. Se o destinatário (dono da verdadeira chave pública) não conseguir decifrar ou obter uma mensagem diferente da original, o ataque é bem sucedido.

$$K_A = e(H_1(ID_B), sP)^{r_A} e(sH_1(ID_A), r_B P) = K$$

$$K'_A = e(H_2(H_1(ID_B)), sP)^{r_A} e(sH_2(H_1(ID_A)), r_B P) = K'$$

Cálculo de L_A e L'_A:

$$L_A = e(H_1(ID_B), sP)^{x_A} e(sH_1(ID_A), x_B P) = L$$

$$L'_A = e(H_2(H_1(ID_B)), sP)^{x_A} e(sH_2(H_1(ID_A)), x_B P) = L'$$

Cálculo de N_A e N'_A:

$$N_A = e(H_1(ID_B), sH_1(ID_A)) = N$$

$$N'_A = e(H_2(H_1(ID_B)), sH_2(H_1(ID_A))) = N'$$

O usuário B irá efetuar os cálculos de maneira similar, e obterá K_B, K'_B, L_B e L'_B, N_B e N'_B. A chave da sessão é computada por:

$$SK = H(A, B, E_A, E_B, r_a r_B P, x_a x_B P, r_a x_B P, x_a r_B P, K, K', L, L', N, N')$$

Os K, L e N são usados na prova para encapsular a entrada para o desafio de CBDH na sessão de teste. Cada um desses valores é necessário para se defender contra uma possível estratégia de ataque do adversário.

O K' é o produto de duas chaves de sessão de Boneh e Franklin (2001) encapsuladas, L' é similar, mas com chaves de longo prazo sem certificado. O N' é o esquema não-interativo de acordo de chaves baseada em identidade proposto por Sakai *et al.* (2000). Os K', L' e N' são necessários para responder às consultas reveladas ao adversário de forma consistente.

5.3.4 Considerações de eficiência

Embora o protocolo seja de uma rodada, a sobrecarga computacional imposta às partes é bastante elevada: cada participante necessita calcular 5 exponenciações em $\mathbb{G}$ e 10 emparelhamentos. Note que se necessita da função de hash H_2 na prova para a plena segurança bilinear Diffie-Hellman computacional.

Se a hipótese do "Gap" bilinear Diffie-Hellman é utilizada, a função de hash H_3 pode ser omitida, economizando duas consultas ao hash e reduz a complexidade do protocolo para 3 exponenciações em $\mathbb{G}$ e 5 computações de emparelhamento (porque K', L', e N' não necessitam ser computados) (Kudla e Paterson, 2005, sec.3.2 pág.7).

Se houver várias execuções do protocolo entre os mesmos usuários (por exemplo, para renegociação de chaves em VPNs), então a complexidade pode ser reduzida armazenando $x_A x_B P$, L, L', N, e N' em memória segura que então reduz a complexidade de execuções sucessivas para 4 exponenciações e 4 cálculos de emparelhamento (ou 2 exponenciações e 2 computações de emparelhamento se a hipótese do "Gap" bilinear Diffie-Hellman for adotada).

A seguir, iremos apresentar um protocolo com as adaptações, supostas pelas considerações de eficiência.

5.4 Descrição do protocolo de Goya-Okida-Terada

Nesta seção, apresentamos o protocolo proposto por Goya *et al.* (2010), que pode ser provado seguro e sob o modelo de segurança forte, com a hipótese BDH e sob o modelo de oráculo aleatório.

É semelhante ao protocolo de Lippold *et al.* (2009), mas combina valores efêmeros com chaves secretas baseada em identidade de maneira diferente.

Essa combinação foi introduzida por Shim (2003) num acordo de chave baseado em identidade, que foi quebrado.

Em Chen *et al.* (2007) e em Huang e Cao (2009b) as variações no protocolo de Shim foram provados seguros.

5.4.1 Instalação

Seja G e G_T grupos de ordem prima q com $P \in G$ como gerador de G.

Seja $e : G \times G \rightarrow G_T$ um emparelhamento bilinear admissível.

O KGC escolhe aleatoriamente $s \in_R [1, q]$ como uma chave mestra e define sua chave pública como sP.

Para um parâmetro de segurança n, o KGC seleciona três funções de hash criptográficas.

$H : \{0,1\}^* \rightarrow \{0,1\}^n$

$H_1 : \{0,1\}^* \rightarrow G$

$H_2 : G \rightarrow G$

O KGC publica os parâmetros $= \langle q, G, G_T, e, k, P, sP, H, H_1, H_2 \rangle$.

5.4.2 Chaves do usuário

Cada usuário U:

- tem um único $ID_U \in \{0,1\}^*$ e valores públicos $(Q_{1_U}, Q_{2_U}) \in G^2$, onde $Q_{1_U} = H_1(ID_U)$ e $Q_{2_U} = H_2(Q_{1_U})$
- escolhe um valor secreto $x_U \in_R [1, q]$
- calcula sua chave pública $x_U P \in G$
- de forma segura recebe do KGC seus segredos parciais $(d_{1_U}, d_{2_U}) \in G^2$, onde $d_{1_U} = sQ_{1_U}$ e $d_{2_U} = sQ_{2_U}$ (ambos são calculados exclusivamente pelo KGC com sua chave mestre).

5.4.3 Troca de mensagens

Para estabelecer uma chave compartilhada, os usuários A e B escolhem aleatoriamente suas chaves parciais efêmeras em $[1, q]$, respectivamente denotados por r_A e r_B, e calculam $r_A P$ e $r_B P$.

Eles trocam as seguintes mensagens:

$$A \rightarrow B : E_A = (r_A P, x_A P)$$

$$B \rightarrow A : E_B = (r_B P, x_B P)$$

5.4.4 Cálculo das chaves

Para dois participantes A e B:

1. Ao receber E_B de B, A verifica se ele está em G^2. Se a verificação for positiva, A calcula

   $K_1 = e(r_B P + Q_{1_B}, r_A s P + d_{1_A})$

   $K_2 = e(r_B P + Q_{2_B}, r_A s P + d_{2_A})$

   $L_1 = e(Q_{1_B}, sP)^{x_A} \cdot e(d_{1_A}, x_B P)$

   $L_2 = e(Q_{2_B}, sP)^{x_A} \cdot e(d_{2_A}, x_B P)$

   $N_1 = e(Q_{1_B}, d_{1_A})$

   $N_2 = e(Q_{2_B}, d_{2_A})$

   E, então A calcula $M = (x_A(x_B P), r_A(r_B P), x_A(r_B P), r_A(x_B P))$ e a chave de sessão como $SK = H(A, B, E_A, E_B, M, K_1, K_2, L_1, L_2, N_1, N_2)$.

2. Da mesma forma, ao receber E_A de A, B verifica se ele está em G^2. Se a verificação for positiva, B computa

   $K_1 = e(r_A P + Q_{1_A}, r_B s P + d_{B_1})$

   $K_2 = e(r_A P + Q_{2_A}, r_B s P + d_{B_2})$

   $L_1 = e(Q_{1_A}, sP)^{x_B} \cdot e(d_{B_1}, x_A P)$

   $L_2 = e(Q_{2_A}, sP)^{x_B} \cdot e(d_{B_2}, x_A P)$

   $N_1 = e(Q_{1_A}, d_{B_1})$

   $N_2 = e(Q_{2_A}, d_{B_2})$

   Então B calcula $M = (x_B(x_A P), r_B(r_A P), r_B(x_A P), x_B(r_A P))$ e a mesma chave de sessão.

5.4.5 Considerações de eficiência

Em comparação com o protocolo de Lippold *et al.* (2009) (LBG), a essa proposta CL-AKA tem menos operações de emparelhamento bilinear, que é a operação mais custosa dentre as executadas pelos protocolos.

Se usarmos uma hipótese forte, o *Gap* BDH, os valores K_2, L_2, N_2 podem ser descartados de ambos os protocolos.

Neste caso o protocolo de Goya *et al.* (2010) (GOT) calcula 4 emparelhamentos, em contra partida ao LBG que calcula 5 emparelhamentos.

Se houver várias execuções entre os mesmos usuários, então os valores L_1, L_2, N_1, N_2 e $x_A x_B P$ podem ser pré-computados e armazenados em memória segura.

A combinação da hipótese *Gap* com o armazenamento seguro dos valores pré-computados, o protocolo de GOT exige que cada usuário calcule 1 emparelhamento, contra o protocolo de LBG, que exige 1 emparelhamento e 1 exponenciação em G_T, as duas operações mais caras.

A Tabela 5.1 mostra as quantidades de operações necessárias para executar uma sessão por um participante, para ambos os protocolos: LBG e GOT.

Na última linha da tabela contém os resultados experimentais obtidos a partir da implementação em Java da aplicação executado na plataforma Intel T2080 em 1.73GHz.

Neste experimento, foi utilizada uma curva elíptica supersingular $E(\mathbb{F}_p) : y^2 = x^3 + x$, definida sobre um corpo com 256 bits de característica prima, com o grau de mergulho $k = 2$.

	Hipótese BDH		Hipótese Gap BDH		Gap BDH + Armazenamento	
	LBG	GOT	LBG	GOT	LBG	GOT
Emparelhamentos	10	8	5	4	1	1
Exponenciações em G_T	0	0	0	0	1	0
Multiplicações em G_T	4	2	2	1	1	0
Multiplicações em G	7	6	5	5	4	5
Adições em G	0	4	0	2	0	2
Tempo (s)	29.62	23.65	15.66	13.07	6.72	5.22

Tabela 5.1: *Comparação dos protocolos*

Embora ambos os protocolos ainda serem muito caros, GOT apresenta uma melhoria.

O CL-AKA de GOT é cerca de 20,2% mais rápido do que LBG, sob hipótese BDH; 16,5% mais rápido sob hipótese Gap BDH e 22,3% mais rápido com o armazenamento e a hipótese Gap BDH.

5.5 Protocolo proposto

Um problema que detectamos nessa classe de protocolos é a carência de opções eficientes, que levam em conta a reduzida capacidade de memória dos dispositivos móveis e que ao mesmo tempo ofereçam elevado nível de segurança.

Para um nível de segurança de 128 bits, os emparelhamentos em curvas supersingulares necessitam de um corpo base de 1083 bits. Já os emparelhamentos sobre curvas BN o corpo base é de apenas 128 bits.

Porém, os protocolos, anteriormente, estudados são protocolos baseados em emparelhamentos do Tipo 1, ou seja, emparelhamentos simétricos.

Felizmente, o trabalho pioneiro de Dupont e Enge (2002) mostrou que é possível transformar um protocolo que utiliza emparelhamentos simétricos em outro para utilizar emparelhamentos assimétricos. Neste trabalho, os autores descrevem que a solução basicamente, será duplicar a quantidade de parâmetros compartilhados no sistema.

Isso significa que como anteriormente tinha-se um $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ com $P, Q \in \mathbb{G}$, agora tem-se $P \in \mathbb{G}_2$.

Portanto todos os parâmetros que eram utilizados no primeiro grupo $\mathbb{G}_1$ deverão estar no segundo $\mathbb{G}_2$, pois queremos utilizar um a função de hash no grupo $\mathbb{G}_1$ que é mais eficiente do que no grupo $\mathbb{G}_2$.

A primeira vista, tal contexto pode parecer não apresentar nenhum ganho significativo, mas baseado-se nos esquemas de acordo de chaves práticos, como o utilizado no SSLv.3, por exemplo (Oppliger, 2009).

E considerar que apenas um dos usuários iniciará o acordo. Então o ambiente será como descrito:

5.5.1 Instalação

Seja G_1, G_2 e G_T grupos de ordem prima q com $P \in G_2$ gerador de G_2.

Seja $e : G_1 \times G_2 \rightarrow G_T$ um emparelhamento bilinear admissível.

O KGC escolhe aleatoriamente $s \in_R [1, q]$ como uma chave mestra e, para fazer as adaptações, definem-se suas duas chaves públicas como sP.

Para um parâmetro de segurança k, o KGC seleciona três funções de hash criptográficas.

$H : \{0,1\}^* \rightarrow \{0,1\}^k$

$H_1 : \{0,1\}^* \rightarrow G_1$

$H_2 : \{0,1\}^* \rightarrow G_2$

$H_3 : G_1 \rightarrow G_1$

$H_4 : G_2 \rightarrow G_2$

O KGC publica os parâmetros $= \langle q, G_1, G_2, G_T, e, k, P, sP, H, H_1, H_2, H_3, H_4 \rangle$.

5.5.2 Chaves do usuário

Cada usuário U:

- tem um único $ID_U \in \{0,1\}^*$ e valores públicos $(Q_{1_U}, Q_{2_U}, R_{1_U}, R_{2_U}) \in G^2$, onde $Q_{1_U} = H_1(ID_U)$, $Q_{2_U} = H_2(Q_{1_U})$, $R_{1_U} = H_3(ID_U)$ e $R_{2_U} = H_4(ID_U)$
- escolhe um valor secreto $x_U \in_R [1, q]$
- calcula sua chave pública $x_U P_2 \in G$
- de forma segura recebe do KGC seus segredos parciais $(d_{1_U}, d_{2_U}) \in G^2$, onde $d_{1_U} = sQ_{1_U}$ e $d_{2_U} = sQ_{2_U}$ e $(\sigma_{1_U}, \sigma_{2_U}) \in G^2$, onde $\sigma_{1_U} = R_{1_U}$ e $\sigma_{2_U} = sR_{2_U}$ (ambos são calculados exclusivamente pelo KGC com sua chave mestre).

5.5.3 Troca de mensagens

Para estabelecer uma chave compartilhada, os usuários A e B escolhem aleatoriamente suas chaves parciais efêmeras em $[1, q]$, respectivamente denotados por r_A e r_B, e calculam $r_A P$ e $r_B P$.

Eles trocam as seguintes mensagens:

$$A \rightarrow B : E_A = (r_A P, x_A P)$$

$$B \rightarrow A : E_B = (r_B P, x_B P)$$

5.5.4 Cálculo das chaves

Para dois participantes A e B:

1. Ao receber E_B de B, A verifica se ele está em G_2^2. Se a verificação for positiva, A calcula

   $K_1 = e(Q_{1_B}, sP)^{r_A} \cdot e(d_{1_A}, r_B P)$

   $L_1 = e(Q_{1_B}, sP)^{x_A} \cdot e(d_{1_A}, x_B P)$

   $N_1 = e(Q_{1_B}, d_{2_A})$

   E, então A calcula $M = (x_A(x_B P), r_A(r_B P), x_A(r_B P), r_A(x_B P))$ e a chave de sessão como $SK = H(A, B, E_A, E_B, M, K_1, L_1, N_1)$.

2. Da mesma forma, ao receber E_A de A, B verifica se ele está em G^2. Se a verificação for positiva, B computa

   $K_1 = e(d_{1_B}, r_A P) \cdot e(r_B Q_{1_A}, sP)$

   $L_1 = e(Q_{1_A}, sP)^{x_B} \cdot e(d_{B_1}, x_A P)$

   $N_1 = e(d_{B_1}, Q_{A_2})$

   Então B calcula $M = (x_B(x_A P), r_B(r_A P), r_B(x_A P), x_B(r_A P))$ e a mesma chave de sessão.

5.5.5 Considerações de eficiência

Nesta nova abordagem, o protocolo proposto apresenta o comportamento similar ao protocolo GOT, porém ao combinar a hipótese *Gap* com o armazenamento em memória segura, o protocolo exige que cada usuário calcule 1 emparelhamento e 1 exponenciação em G_T, a segunda operação mais cara.

Mas esse gasto é compensado pela diminuição do tempo de computação das funções de hash em $\mathbb{G}_2$ e o nível de segurança fornecido pelas curvas elípticas ordinárias.

5.6 Resumo

Neste capítulo foram apresentados os protocolos de Lippold-Boyd-González, bem como sua arquitetura, o protocolo de Goya-Okida-Terada e uma modificação do protocolo de Lippold-Boyd-González para utilização de emparelhamentos assimétricos.

Capítulo 6

Implementação

Neste capítulo, vamos apresentar a descrição das técnicas utilizadas para a implementação de protocolos criptográficos sem certificado. Essas técnicas podem ser divididas em vários níveis, ilustrados na Figura 6.1. Os níveis mais baixos são compostos por algoritmos para a aritmética de corpos finitos; que basicamente envolve soma, subtração, multiplicação e inversão em $\mathbb{F}_{q^m}$.

Acima deles, tem-se a aritmética de curvas elípticas, que consiste basicamente na adição de pontos e na multiplicação escalar de ponto (adição repetida). Acima, tem-se o nível do cálculo de emparelhamentos, cujo algoritmo principal é o algoritmo de Miller. No topo estão os protocolos utilizados.

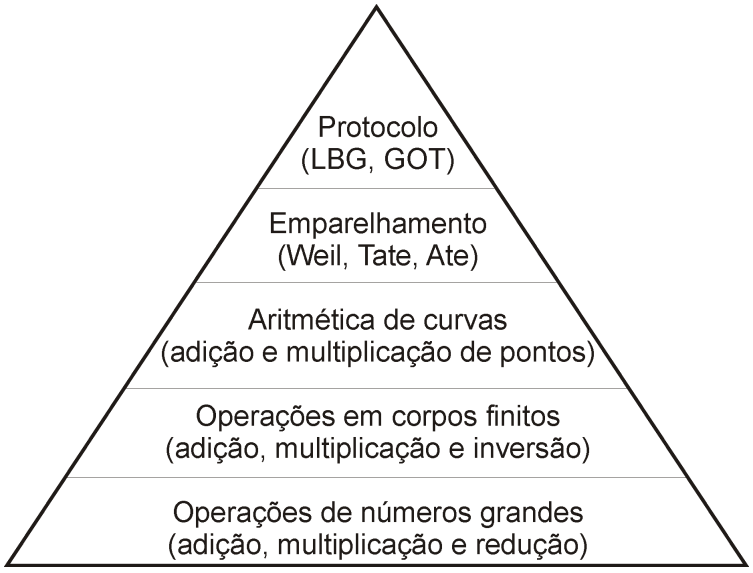


Figura 6.1: *Modelo da implementação da biblioteca criptográfica*

6.1 Operações de precisão arbitrária

A adoção da linguagem Java, é um pré-requisito do Projeto, viabiliza a execução em diferentes plataformas e em diversos tipos de dispositivos móveis.

Muitos algoritmos criptográficos dependem do uso de inteiros grandes e aritmética de redução modular.

Felizmente a linguagem Java fornece tudo isso na classe `java.math.BigInteger`, que representa os inteiros de precisão arbitrária, otimizada para obter velocidade.

A classe BigInteger suporta inteiros de precisão arbitrária. Isto significa que pode-se representar precisamente valores integrais de qualquer tamanho sem perder qualquer informação durante as operações.

Se p é um primo, o módulo do corpo finito e o comprimento de bit de p é o comprimento da chave, a classe BigInteger fornece um construtor que faz exatamente o que queremos:

`public BigInteger(int bitLength, int certainty, Random rnd)`

`bitLength` é o comprimento desejado para o novo BigInteger. O parâmetro `certainty` determina a probabilidade de o número ser primo. Em particular, para um parâmetro de segurança n, a probabilidade de um número ser primo é $1 - 0,5n$. Esse construtor simplesmente cria um número com o comprimento de bits dado, usando o `Random` fornecido.

O bit de ordem elevada será sempre definido, garantindo que o comprimento de bit do novo número coincida com o comprimento de bit solicitado.

A classe possui métodos que implementam a maioria das operações matemáticas. Os mais comuns são `add()`,`subtract()`,`multiply()`, e `divide()`. Muitas outras operações úteis também são encapsulados por métodos, inclusive o método `modPow()` que precisamente atende a necessidade das implementações.

6.1.1 Método de multiplicação de Karatsuba

O método de Karatsuba é utilizado para acelerar os algoritmos de emparelhamento de Ate.

O algoritmo de Karatsuba resolve o problema em $\Theta(n \lg 3)$ unidades de tempo. (Observe que $\lg 3$ fica entre $1,5$ e $1,6$.)

A função $n \lg 3$ é solução da recorrência $\Theta(n) = 3T(\lfloor n/2 \rfloor) + n$.

Sejam u e v dois números com no máximo n dígitos cada. Suponha, por enquanto, que n é par.

Seja p o número formado pelos $n/2$ dígitos mais significativos de u e seja q o número formado pelos $n/2$ dígitos menos significativos de u.

Assim,

$$u = p \cdot 10^{n/2} + q.$$

Defina r e s analogamente para v, de modo que $v = r10^{n/2} + s$. Temos então

$$uv = pr \cdot 10^n + (ps + qr) \cdot 10^{n/2} + qs. \tag{6.1}$$

Esta expressão reduz a multiplicação de dois números com no máximo n dígitos cada a quatro multiplicações (a saber, p por r, p por s, q por r e q por s) de números com no máximo $n/2$ dígitos cada.

Infelizmente, essa redução não é suficiente para tornar a multiplicação mais eficiente.

Agora observe que os três números de que se precisa do lado direito da equação 6.1 – a saber pr, $(ps + qr)$ e qs – podem ser obtidos com apenas três multiplicações, pois

$$ps + qr = \mathbf{y} - pr - qs, \text{ sendo } \mathbf{y} = (p + q)(r + s),$$

e, portanto a equação 6.1 pode ser substituída por

$$uv = pr \cdot 10^n + (\mathbf{y} - pr - qs)10^{n/2} + qs.$$

Apesar de envolver duas adições e duas subtrações adicionais, essa operações consomem muito menos tempo que as multiplicações.

Se n não é par, basta trocar $n/2$ por $k = \lfloor n/2 \rfloor$. Tem-se então $u = p \cdot 10^k + q$ e $v = r \cdot 10^k + s$ e portanto

$$uv = pr \cdot 10^{2k} + (\mathbf{y} - pr - qs)10^k + qs.$$

Esta é a base do algoritmo.

6.2 Operações em corpos finitos

Em álgebra abstrata, o corpo $\mathbb{F}_{p^2}$ pode ser representado como $\mathbb{F}_{p^2} = \{a + b\alpha | a, b \in \mathbb{Z}_p\}$, assim seus elementos podem ser representados:

$$\{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\}$$

$$\begin{aligned}(a_1 + b_1\alpha) + (a_2 + b_2\alpha) &= (a_1 + a_2) + (b_1 + b_2)\alpha \\ (a_1 + b_1\alpha).(a_2 + b_2\alpha) &= a_1a_2 + a_1b_2\alpha + b_1a_2\alpha + b_1b_2\alpha^2\end{aligned}$$

Por isso, as operações no corpo $\mathbb{F}_{p^2}$ recaem sobre as operações dos números complexos.

O corpo de extensão $\mathbb{F}_{p^{12}}$ é representado usando torres de extensões:

$$\begin{aligned}\mathbb{F}_{p^2} &= \mathbb{F}_p[u]/(u^2 + 2), \\ \mathbb{F}_{p^6} &= \mathbb{F}_{p^2}[v]/(v^3 - \xi)\end{aligned}$$

onde $\xi = -u - 1$, e $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^6}[w]/(w^2 - v)$.

Temos também a representação $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^2}[W]/(W^6 - \xi)$, onde $W = w$.

Portanto, um elemento $\alpha \in \mathbb{F}_{p^{12}}$ pode ser representado em qualquer uma das três maneiras:

$$\begin{aligned}\alpha &= a_0 + a_1w, \text{ onde } a_0, a_1 \in \mathbb{F}_{p^6} \\ &= (a_{0,0} + a_{0,1}v + a_{0,2}v^2) + (a_{1,0} + a_{1,1}v + a_{1,2}v^2)w, \text{ onde } a_{i,j} \in \mathbb{F}_{p^2} \\ &= a_{0,0} + a_{1,0}W + a_{0,1}W^2 + a_{1,1}W^3 + a_{0,2}W^4 + a_{1,2}W^5.\end{aligned}$$

Se $(\mathsf{m}, \mathsf{s}, \mathsf{i})$, $(\tilde{\mathsf{m}}, \tilde{\mathsf{s}}, \tilde{\mathsf{i}})$, $(\mathsf{M}, \mathsf{S}, \mathsf{I})$ indicam o custo de multiplicação, quadratura, inversão em $\mathbb{F}_p$, $\mathbb{F}_{p^2}$, $\mathbb{F}_{p^{12}}$, respectivamente.

Experimentalmente, temos $\mathsf{s} \approx 0,9\mathsf{m}$ e $\mathsf{i} \approx 41\mathsf{m}$. Em nossas estimativas de custo que se seguem, vamos adotar a hipótese para simplificar $\mathsf{s} \approx \mathsf{m}$.

Se $a \in \mathbb{F}_p$ e $\alpha \in \mathbb{F}_{p^n}$ para $n \in \{2, 6, 12\}$, então o custo da computação $a \cdot \alpha$ é um $n\mathsf{m}$.

Para aritmética $\mathbb{F}_{p^2}$, tem-se $\tilde{\mathsf{m}} \approx 3\mathsf{m}$ (usando o método Karatsuba, que reduz a multiplicação em uma extensão quadrática a 3 (ao invés de 4) multiplicações no corpo menor), $\tilde{\mathsf{s}} \approx 2\mathsf{m}$ (usando o método complexo de: $(a + bu)^2 = (a - b)(a + 2b) - ab + (2ab)u$), e $\tilde{\mathsf{i}} \approx \mathsf{i} + 2\mathsf{m} + 2\mathsf{s}$ (uma vez que $(a + bu)^{-1} = (a - bu)/(a^2 + 2b^2)$).

Note também que a p-ésima potência é livre de multiplicação em $\mathbb{F}_{p^2}$, uma vez que $(a+bu)^p = a-bu$.

O método Karatsuba reduz uma multiplicação em uma extensão cúbica a 6 (invés de 9) multiplicações no corpo menor. Consequentemente, uma multiplicação em $\mathbb{F}_{p^6}$ custa $18\mathsf{m}$.

A quadratura em $\mathbb{F}_{p^6}$ custa $2\tilde{\mathsf{m}}+3\tilde{\mathsf{s}} = 12\mathsf{m}$ via as seguintes fórmulas: se $\beta = b_0+b_1v+b_2v^2 \in \mathbb{F}_{p^6}$, onde $b_i \in \mathbb{F}_{p^2}$, então $\beta_2 = (A+D\xi)+(B+E\xi)v+(B+C+D-A-E)v^2$ onde $A = b_0^2$, $B = 2b_0b_1$, $C = (b_0-b_1+b_2)^2$, $D = 2b_1b_2$ e $E = b_2^2$ (Chung e Hasan, 2007).

Finalmente, como mostrado em Scott (2007) (Seção 3.2), a inversão em $\mathbb{F}_{p^6}$ pode ser reduzida a uma inversão, 9 multiplicações e 3 quadraturas em $\mathbb{F}_{p^2}$.

Uma vez que $\mathbb{F}_{p^{12}}$ é uma torre quadrática, cúbica e de extensões quadráticas, o método Karatsuba dá $\mathsf{M} \approx 54\mathsf{m}$.

Usando o método complexo para quadratura em $\mathbb{F}_{p^{12}}$ e Karatsuba para a multiplicação em $\mathbb{F}_{p^6}$ e $\mathbb{F}_{p^2}$, temos $\mathsf{S} \approx 36\mathsf{m}$.

Note, no entanto, que, se $\alpha^2 = a+bw \in \mathbb{F}_{p^{12}}$ satisfaz $\alpha^{p^6+1} = 1$ (e, portanto, $a^2-b^2v = 1$), então tem-se $\alpha^2 = (a+bw)^2 = (a^2+b^2v)+(2ab)w = (2b^2v+1)+[(a+b)^2-b^2-b^2v-1]w$.

Consequentemente a quadratura com α, uma operação denotada por S', pode ser reduzida para 2 quadraturas em $\mathbb{F}_{p^6}$ e assim $\mathsf{S}' \approx 24\mathsf{m}$ (Stam e Lenstra, 2003).

Invertendo α é essencialmente livre, porque $\alpha^{-1} = \alpha^{p^6}$.

Uma vez que a inversão em $\mathbb{F}_{p^{12}}$ pode ser reduzida a 1 inversão, 2 multiplicações, e 2 quadraturas em $\mathbb{F}_{p^6}$, segue-se que $\mathsf{I} \approx \mathsf{i}+97\mathsf{m}$.

E_3 tem uma torção de grau 6 sobre $\mathbb{F}_{p^2}$, ou seja, $E/\mathbb{F}_{p^2} : y^2 = x^3+3/\xi$.

O monomorfismo $\phi_6 : \mathbb{G}_2' \to \mathbb{G}_2$ é dado por $(x,y) \mapsto (xW^2, yW^3)$.

6.3 Operações em curvas elípticas de característica prima

De mesmo modo, a aritmética de curva elíptica, introduzida no Capítulo 3 (pág.15), será revisada para o caso específico de curvas elípticas de característica prima e na Seção 6.5 (pág.65) quando apresentamos a implementação do emparelhamento Ate sobre uma curva BN.

6.4 Curvas para calcular emparelhamentos

Uma típica curva elíptica $E/\mathbb{F}_q$ fornece uma estrutura inadequada para o cálculo de um emparelhamento, porque o grau de mergulho dos subgrupos de $E(\mathbb{F}_q)$ de ordem prima grande é normalmente muito alto para fazer o cálculo de um emparelhamento prático.

Para fornecer uma estrutura adequada a implementação de algoritmos baseados em emparelhamento, precisam das seguintes propriedades:

- A existência de um subgrupo de $E(\mathbb{F}_q)$ da ordem prima grande p.
- Um baixo grau de mergulho de $E(\mathbb{F}_q)$.

A primeira dessas condições é cuidadosamente definida: os parâmetros de segurança desejados de um sistema irão determinar a ordem necessária do G_1.

No entanto, definir um grau de mergulho baixo requer a criação de um limite arbitrário.

Embora, um grau de mergulho $k < (\log q)^2$ seja baixo suficiente para fazer o cálculo do logaritmo discreto em $\mathbb{F}_{q^k}$ eficiente no sentido teórico, um subgrupo com tal grau de mergulho ainda fornece uma estrutura inviável para a implementação de um emparelhamento.

A exigência mais prática é que o grau de mergulho de G_1 em relação ao p seja menor que $(\log_2 p)/8$, onde a constante $(\log_2 p)/8$ é escolhida de modo arbitrário.

Isso fornece a motivação para a seguinte definição.

Definição 6.1. Uma curva elíptica E/q possui *emparelhamento amigável*, se

- Existe um subgrupo de $E(\mathbb{F}_q)$ de uma ordem prima p suficientemente grande.
- O grau de mergulho de $E(\mathbb{F}_q)$ com relação a p é menor que $(\log_2 p)/8$.

Note que, se $E/\mathbb{F}_q$ é supersingular e $E(\mathbb{F}_q)$ tem um subgrupo de ordem adequada então $E/\mathbb{F}_q$ é automaticamente um emparelhamento amigável, porque deve existir um $k \leq 6$ para o grau de mergulho de $E(\mathbb{F}_q)$.

Encontrar emparelhamento amigável em curvas ordinárias, por outro lado, é uma área ativa de pesquisa.

Mas um grande progresso tem sido feito para fornecer curvas capazes de permitir a implementação eficiente de criptografia baseada em emparelhamentos, em vários níveis de segurança.

Entre essas alternativas, algumas escolhas são mais eficientes que outras, no entanto.

As técnicas existentes para a geração de emparelhamentos amigáveis em curvas ordinárias usam uma variante da técnica de grupos geradores de curva elíptica de ordem conhecida que é chamado de algoritmo de multiplicação complexa (MC).

A geração de curvas elípticas adequadas usando o algoritmo MC é baseada na seguinte propriedade, em que o inteiro δ é o discriminante MC da curva resultante e o inteiro t representa o seu traço (1363-2000, 2000).

Propriedade 6.1. *(Atkin e Morain, 1993)* Seja q um primo ímpar de tal forma que $4q = t^2 + \delta V^2$ para inteiros V, t, e δ. Então, há uma curva elíptica $E/\mathbb{F}_q$ com $\#E(\mathbb{F}_q) = q + 1 - t$.

Uma curva elíptica pode ser construída com essas propriedades, se e somente se mantidas as seguintes condições, nenhuma das quais representam problema para a geração de curvas ordinárias para utilização em algoritmos baseados em emparelhamento (Lay e Zimmer, 1994).

- q é um primo ou potência de primo;
- p é um primo;
- p divide $q + 1 - t$
- $p|(q^k - 1)$ mas $p \not| (q^i - 1)$ para $1 \geq i \geq k$;
- $4q = t^2 + \delta V^2$ para δ e V inteiros.

A estratégia geral para encontrar curvas ordinárias de emparelhamento amigável tem os seguintes passos, os detalhes variam dependendo do grau de mergulho exigido.

- Fixar um grau de mergulho k e encontrar inteiros t, p, q tais que $E/\mathbb{F}_q$ tem traço t, $E(\mathbb{F}_q)$ tenha um subgrupo de ordem prima grande p e grau de mergulho k.

- Usar o algoritmo CM para encontrar a forma explícita de $E/\mathbb{F}_p$ (Washington, 2008, cap. 10).

Os detalhes dos algoritmos para encontrar as curvas com grau de mergulho particular podem ser encontrados em Barreto *et al.* (2002); Miyaji *et al.* (2001), e estão fora do escopo deste trabalho.

A Tabela 6.1 lista os tipos de curvas ordinárias criadas usando essa estratégia que provaram ser especialmente úteis para atingir os níveis de segurança padrão, permitindo implementações relativamente eficiente.

Tipo de construção	Grau de mergulho	Referência
MNT (Miyaji, Nakabayashi e Takano)	$k = 3, 4, 6$	Miyaji *et al.* (2001)
Freeman	$k = 10$	Freeman (2006)
BN (Barreto-Naehrig)	$k = 12$	Barreto *et al.* (2005)
BW (Brezing-Weng)	18 $\not\mid k$	Brezing e Weng (2005)

Tabela 6.1: *Tipos úteis de curvas ordinárias*

6.4.1 Eficiência relativa dos parâmetros das curvas para emparelhamento amigável

Um parâmetro geralmente usado para comparar a eficiência relativa dos parâmetros de um algoritmo baseado em emparelhamentos é o tamanho do corpo finito $\mathbb{F}_q$ em relação ao p, o tamanho do subgrupo de ordem prima.

Este parâmetro é denotado por ρ, talvez, para a eficiência *relativa*, e é definido como sendo

$$\rho = \frac{\log q}{\log p}$$

de modo que

$$\frac{\log q^k}{\log p} = \rho \cdot k$$

Em geral, as escolhas de parâmetros com valores pequenos de ρ proporcionam operações de curvas elípticas mais rápidas e, portanto, pode ser preferível escolher parâmetros com valores de ρ maiores.

Embora isso não deva ser tomado como um princípio rígido de desenvolvimento, pois outros "trade-offs" podem ser mais importantes que a velocidade adicional oferecida pelas operações de curvas elípticas mais rápidas.

Na prática, por exemplo, se ρ é próximo de 1, pode ser difícil encontrar um valor de p que é um primo de Solinas. A velocidade adicional no cálculo do emparelhamento (por p ser um primo Solinas) pode mais do que compensar as operações de elíptica curva mais lentas, que são necessárias para um valor maior de ρ.

Para $\rho = 1$, os valores ideais de p, q, k que podem ser utilizados para atingir os níveis de segurança padrão são apresentados na Tabela 6.2.

A pesquisa atual, porém, ainda não encontrou curvas que atendam a todas as exigências da Tabela 6.2. O ajuste mais próximo dos valores ideais que atualmente é possível por meio de curvas conhecidas é mostrada na Tabela 6.3.

Tamanho em bits	Tamanho de p	Grau de mergulho k	Tamanho de q^k
80	171	6	1096
112	228	9	2052
128	256	12	3072
192	384	20	7680
256	512	30	15360

Tabela 6.2: *Parâmetros ideais em função do nível de segurança para $\rho = 1$ usando curvas ordinárias*

Tamanho em bits	Tamanho de p	ρ	Grau de mergulho k	Tamanho de q^k	Construção
80	171	1	6	1096	
112	228	1	9	2052	
128	256	1	12	3072	
192	384	10/9	20	7680	
256	512	15/14	30	15360	

Tabela 6.3: *Melhores parâmetros conhecidos em função do nível de segurança usando curvas ordinárias*

6.5 Implementação de emparelhamentos assimétricos

Por causa da necessidade de atingir nível de segurança de 128 bits, decidimos em adotar as curvas BN de emparelhamento amigável.

Seja E uma curva elíptica ordinária definida sobre $\mathbb{F}_q$ e grau de mergulho k em relação a um divisor primo r de $\#E(\mathbb{F}_q)$.

Suponha ainda que $r^3 \not| \#E(\mathbb{F}_{q^k})$ e $r^2 \not| q^k - 1$.

Seja $P \in E(\mathbb{F}_q)$ um ponto de ordem r, seja $\mathbb{G}_1 = \langle P \rangle$, e μ_r denota o subgrupo de ordem r de $\mathbb{F}^*_{q^k}$.

Suponha que E admite uma torção E' de grau d sobre F_{q^e}, onde $e = k/d$.

Seja E' uma torção para a qual $r|\#E(\mathbb{F}_{q^e})$, a existência de E' é garantida pelo Teorema 9 de Hess *et al.* (2006).

Seja $Q' \in E'(\mathbb{F}_{q^e})$ um ponto de ordem r, e seja $\mathbb{G}'_2 = \langle Q' \rangle$. Então, existe um monomorfismo de grupo $\phi_d : \mathbb{G}'_2 \to E(\mathbb{F}_{q^k})$ eficientemente computável tal que $Q = \phi_d(Q') \notin E(\mathbb{F}_q)$.

O grupo $\mathbb{G}_2 = \langle Q \rangle$ é o subgrupo de Traço 0 de $E(\mathbb{F}_{q^k})[r]$.

O emparelhamento assimétrico considerado nesta seção é o emparelhamento Ate $a_r : \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r$

Considerarmos apenas esse emparelhamento para as curvas elípticas Barreto-Naehrig (BN) (Barreto *et al.*, 2005).

Essa curva elíptica E é definida sobre os corpos primos $\mathbb{F}_p$, tem ordem prima $\#E(\mathbb{F}_p)$, e grau de mergulho $k = 12$.

Elas são especialmente muito adequadas para o nível de segurança de 128 bits, pois se p é um primo de 256 bits, então o método rho de Pollard para computação de logaritmos discretos em $E(\mathbb{F}_p)$ tem tempo de execução de aproximadamente 2^{128}, assim como o algoritmo NFS ("*Number field sieve*") para a computação de logaritmos discretos em corpos de extensão $\mathbb{F}_{p^{12}}$.

As curvas BN também admitem torção sêxtica ($d = 6$), o que significa que muitos cálculos podem ser restritas ao corpo $\mathbb{F}_{p^2}$, por trabalhar com os pontos em $\mathbb{G}'_2$ em vez de pontos em $\mathbb{G}_2$.

A curva de BN com que trabalhamos é

$$E_3/\mathbb{F}_p : y^2 = x^3 + 3$$

com o parâmetro BN $z = 6000000000001F2D$ (em hexadecimal) (Devegili *et al.*, 2007).

Para esta escolha do parâmetro BN, $p = 36z^4 + 36z^3 + 24z^2 + 6z + 1$ é um primo de 256 bits com peso de Hamming 87, $r = \#E_3(\mathbb{F}_p) = 36z^4 + 36z^3 + 18z^2 + 6z + 1$ é um primo de 256 bits com peso de Hamming 91, e $t - 1 = p - r = 6z^2 + 1$ é um inteiro de 128 bits com peso de Hamming 28; aqui $t = p + 1 - r$ é o traço de $E_3/\mathbb{F}_p$. Note que $p = 7 \pmod 8$ (onde -2 não é um quadrado modulo p) e $p = 1 \pmod 6$.

6.6 Implementação do Emparelhamento Ate

O emparelhamento Ate $a_r : \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r$, como proposto por Hess *et al.* (2006), é definido como sendo $a_r(Q, P) = f_{t-1,Q}(P)^{(p^{12}-1)/r}$.

Algoritmo 2: Algorítmo de Miller para emparelhamento Ate para curva elíptica $E_3/\mathbb{F}_p$

Entrada: $P \in \mathbb{G}_1$ e $Q \in \mathbb{G}_2$.

Saída: $a_r(Q, P)$.

início

- Escreva $t - 1$ em binário: $t - 1 = \sum_{i=0}^{L-1} t_i 2^i$;
- $T \leftarrow Q$;
- $f \leftarrow 1$;
- **para** $i = L - 2; i \geq 0; i = i - 1$ **faça**
  - Seja ℓ a linha tangente em T;
  - $T \leftarrow 2T$;
  - $f \leftarrow f^2 \cdot \ell(P)$;
  - **se** $t_i = 1$ **então**
    - Seja ℓ a linha secante a T e Q;
    - $T \leftarrow T + Q$;
    - $f \leftarrow f \cdot \ell(P)$;
- Calcule $f^{(p^{12}-1)/r}$ como:;
- $f \leftarrow f^{p^6-1}$ $f \leftarrow f^{p^2-1}$ $a \leftarrow f^{-(6z+5)}$;
- $b \leftarrow a^p$;
- $b \leftarrow a \cdot b$;
- Calcule $f^p; f^{p^2}; f^{p^3}$;
- $f \leftarrow f^{p^3} \cdot [b \cdot (f^p)^2 \cdot f^{p^2}]^{6z^2+1} \cdot b \cdot (f^p \cdot f)^9 \cdot a \cdot f^4$;
- **devolve** f;

fim

Uma vez que $t \approx \sqrt{r}$ para a curva BN, E_3, o número de iterações na operação de Miller é reduzido pela metade.

Um ponto (X, Y, Z) em coordenadas Jacobiana corresponde ao ponto (x, y) em coordenadas

afins com $x = X/Z^2$ e $y = Y/Z^3$.

Em coordenadas Jacobiana as fórmulas para duplicar um ponto $T = (X, Y, Z)$ são $2T = (X_3, Y_3, Z_3)$, onde $X_3 = 9X^4 - 8XY^2$, $Y_3 = (3X^2)(4XY^2 - X_3) - 8Y^4$ e $Z_3 = 2YZ$.

A linha tangente em T, depois de compensar os denominadores, é $\ell(x, y) = Z_3Z^2y - 2Y^2 - 3X^2(Z^2x - X) \in \mathbb{F}_p[x, y]$ (Chatterjee *et al.*, 2005).

As fórmulas de adição e duplicação são realmente aplicadas em pontos em $E(\mathbb{F}_{p^2})$, garantindo assim que a aritmética da curva elíptica está sobre $\mathbb{F}_{p^2}$ (em vez de mais $\mathbb{F}_{p^{12}}$); um ponto Jacobiano $(X, Y, Z) \in E'(\mathbb{F}_{p^2})$ mapeia convenientemente para o ponto Jacobiano $(XW^2, YW^3, Z) \in E(\mathbb{F}_{p^{12}})$.

No passo 2, T é inicializado com o ponto Jacobiano $(xW^2, yW^3, 1)$, onde $Q = (xW^2, yW^3) \in \mathbb{G}_2$.

A duplicação de ponto no passo 5 custa $3\tilde{\mathsf{m}} + 4\tilde{\mathsf{s}}$.

A linha tangente no ponto afim correspondente a $T = (XW^2, YW^3, Z)$ é $\ell(x, y) = Z_3Z^2y - 2Y^2W^3 - 3X^2W(Z^2x - XW^2) \in \mathbb{F}_{p^{12}}[x, y]$, onde $2T = (X_3W^2, Y_3W^3, Z_3)$.

Calculo de $\ell(P)$ e f^2 no passo 6 custa $3\tilde{\mathsf{m}} + \tilde{\mathsf{s}} + 4\mathsf{m}$ e $36\mathsf{m}$, respectivamente.

Note que $\ell(P)$ é da forma $a+bW+cW^3$ com $a, b, c \in \mathbb{F}_{p^2}$, podemos escrever $\ell(P) = a+(b+cv)w$ onde a e $(b + cv)$ são elementos de $\mathbb{F}_{p^6}$.

Isso, segue que o produto de $\ell(P)$ e $f^2 = f_0 + f_1w$ (onde $f_0, f_1 \in \mathbb{F}_{p^6}$) pode ser computado usando a técnica de Karatsuba a um custo de $13\tilde{\mathsf{m}}$.

O custo dos passos 7 a 10 é de $8\tilde{\mathsf{m}} + 3\tilde{\mathsf{s}}$ para calcular $T + Q$, $2\tilde{\mathsf{m}} + 4\mathsf{m}$ para valoração de $\ell(P)$, e $13\tilde{\mathsf{m}}$ para computar $f \cdot \ell(P)$.

Aqui, a linha (após a compensação de denominadores) secante ao ponto afim correspondente a $T = (X_1W^2, Y_1W^3, Z_1)$ e o ponto $Q = (X_2W^2, Y_2W^3)$ é $\ell(x, y) = (y - Y_2W^3)Z_3 - (Y_2Z_1^3 - Y_1)W(x - X_2W^2) \in \mathbb{F}_{p^{12}}[x, y]$, onde $T + Q = (X_3W^2, Y_3W^3, Z_3)$ e $\ell(P)$ é da forma $a + bW + cW^3$ com $a, b, c \in \mathbb{F}_{p^2}$.

Definição de $\mathsf{s} = \mathsf{m}$ produz os custos estimados $15722\mathsf{m}$, $7246\mathsf{m}+\mathsf{i}$, e $22968\mathsf{m}+\mathsf{i}$ para a operação de Miller, a exponenciação final, e Algoritmo 2 (pág.66), respectivamente.

6.7 Hash em $\mathbb{G}_1$ e em $\mathbb{G}_2$

Como foi apresentado no Capítulo 5, são necessárias funções de hash que mapeiem identidades em pontos P de forma que não se saiba escrever $P = kG$ para algum ponto base fixo G.

Primeiramente, pode-se mapear identidades no grupo $E(\mathbb{K})$ de uma curva elíptica mapeando-se a identidade em um valor x de $\mathbb{K}$, e calculando-se $y = \sqrt{x^3 + ax + b}$. Caso a raiz quadrada não exista, pode-se incrementar x e tentar novamente. Em Brier *et al.* (2009); Icart (2009) o processo é mais detalhado, inclusive fornecem-se algoritmos determinísticos para o hash quando $p \equiv 2 \pmod 3$.

Resta apenas um ponto a ser esclarecido, que surge ao se considerar que $\mathbb{G}_1 = E(\mathbb{K})[r]$, isto é, os pontos de r-torção em $E(\mathbb{K})$.

Como r é primo, então $\mathbb{G}_1$ consiste nos pontos de ordem r mais o ponto no infinito. Sabe-se como mapear uma identidade em $E(\mathbb{K})$, mas falta saber como garantir que o ponto tenha ordem r. Seja n a ordem de $E(\mathbb{K})$ e $c = n = r$, também denominado *cofator*.

Então, pode-se garantir que o ponto mapeado tenha ordem r calculando-se $c\dot{H}_1(ID)$.

Em $\mathbb{G}_2$ procede-se de maneira análoga. Se for utilizada curva twist de grau d, então o cofator consiste em $c = n_0 = r$, onde n_0 é a ordem do twist $E'(\mathbb{F}_{q^k} = d)$ que pode ser calculada de acordo com Hess *et al.* (2006) (prop. 2).

Neste caso, é importante notar que o cofator é maior (do tamanho de p se o twist for em $E(\mathbb{F}_{p^2})$, por exemplo), portanto H_2 é mais cara que H_1. Mesmo assim, alguns trabalhos como Scott *et al.* (2009) fornecem técnicas para se melhorar a performance de H_2.

Vale lembrar que, se for utilizado o emparelhamento de Tate, não é necessário multiplicar pelo cofator em $\mathbb{G}_2$ já que qualquer ponto é representante de uma classe lateral.

Os resultados da implementação serão apresentados no próximo capítulo. Então o estudo de caso do Projeto Borboleta será descrito a seguir.

6.8 Estudo de caso – Projeto Borboleta

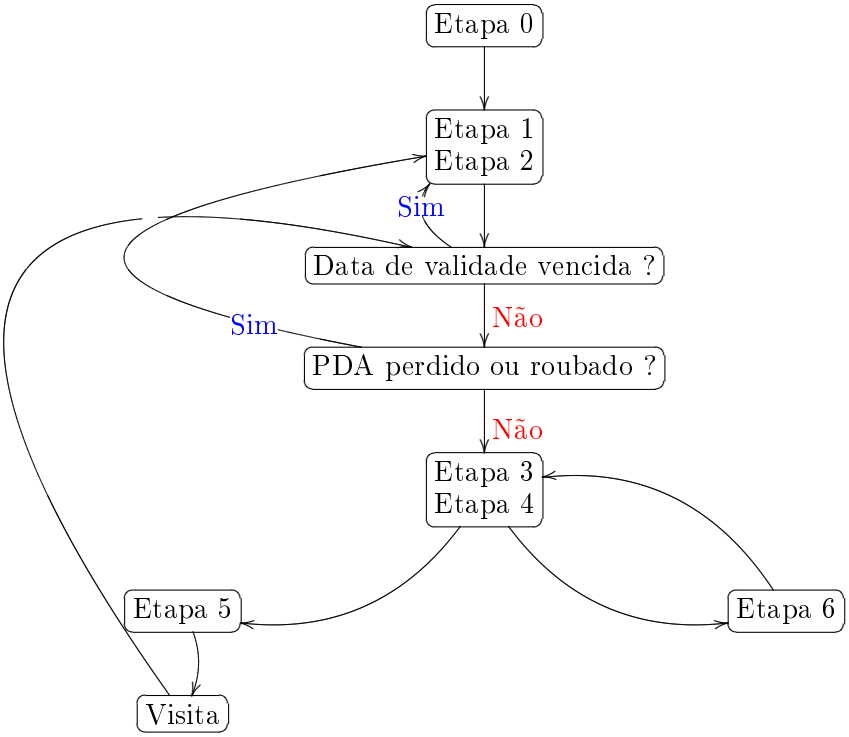
No projeto Borboleta, precisamos especificar quais dados e como serão utilizados no sistema.

Como o sistema está, inicialmente, restrito ao Centro de Saúde Escola Butantã, definimos a identificação de cada usuário U como $ID_U = ID_{\text{do agente de saúde}}||\text{data de validade}$

Além do protocolo implementado, necessitamos de uma metodologia e os procedimentos para cobrir casos de vencimento das chaves de longa duração e nos piores, de perda ou furto do aparelho certificado.

A validade das chaves de longa duração inicialmente foi estimada em uma semana, necessitando de testes em campo para uma estimação mais adequada.

A metodologia para os procedimentos dos casos listados acima, são ilustrados no Esquema 6.8.



0. Inicialmente iremos realizar a instalação das chaves do Borboleta a partir do protocolo proposto como descrito na Seção 5.5.1. Este passo será realizado apenas uma vez, quando o KGC for instalado.

1. A seguir, o dispositivo móvel será inicializado como descrito na Seção 5.5.2, sempre será utilizado canais seguros entre as entidades para evitar clonagens, ou seja, será realizado em uma sala de acesso restrito.

A partir desta etapa, serão realizadas sempre que houver necessidade de renovação dos segredos parciais de longa duração.

2. Em seguida, o sistema do dispositivo móvel trocará os segredos de longa duração baseado em *Identidade*, sempre que houver uma sincronização com o banco de dados dos prontuários médicos, como na Seção 5.5.3.

3. E calcularão as chaves de efêmeras e trocarão os segredos de curta duração como apresentado na Seção 5.5.3.

4. Assim, nesta etapa são calculados os emparelhamentos do protocolo como apresentado na Seção 5.5.4. Como apenas a variável K_U utiliza os segredos efêmeros, apenas este será computado a cada acordo de chaves. Os demais podem ser calculados durante a etapa 2 e armazenado em memória segura.

5. E calcularão as chaves de sessão como apresentado na Seção 5.5.4. Depois dessas operações, o dispositivo móvel está disponível para o agente realizar a visita domiciliar dos pacientes.

6.9 Resumo

Neste capítulo apresentamos como as operações de precisão arbitrária são implementadas na linguagem Java, as operações de corpos finitos, o cálculo de emparelhamentos e a implementação do emparelhamento de Ate.

Capítulo 7

Experimentos

Neste capítulo, vamos apresentar os experimentos e os testes realizados.

Para realizar os experimentos foi montado um protótipo para simular com maior realismo os mecanismos de troca de chaves no sistema.

O protótipo consiste em um dispositivo móvel ligado via cabo USB a um computador, onde os participantes geram suas respectivas chaves secretas e realizam a combinação de chaves. Como podemos observar, os participantes serão o dispositivo móvel e o servidor de banco de dados do Borboleta.

Foram utilizados os seguintes equipamentos:

- Computador
  Fabricante: Acer
  Modelo: Aspire 5610Z
  Processador: Genuine Intel® CPU T2080 1,73GHz
  Memória (RAM): 1,0GB
  Sistema(s) operacional(is): Windows Vista Home Premium 32-bit Service Pack 1
  Ubuntu 9.04
  Dispositivo de rede: Broadcom 440x 10/100 Integrated Controller
  Adaptador de rede wireless: Atheros AR5007EG Wireless Network Adapter

- Celular
  Fabricante: HTC
  Modelo: P3451
  Processador: OMAP™850 201MHz
  Memória (RAM): 128 MB
  Sistema operacional: Windows Mobile® 6 Professional
  Dispositivo de rede: Wi-Fi compatível com IEEE 802.11 b/g

- Compilador
  J2ME Wireless Toolkit 2.5.2 (dispositivo móvel) e J2SE 1.6.0 (computador)
  IDE NetBeans 6.7.1

A curva de BN considerada foi

$$E_3/\mathbb{F}_p : y^2 = x^3 + 3$$

com o parâmetro BN $z = 6000000000001F2D$ (em hexadecimal) como foi citada na Seção 6.5 (pág.65).

Os tempo de execução no computador para as operações envolvidas nessa curva com o protocolo proposto são apresentados na Tabela 7.4.

Operações	Tempo de execução (ms)
Emparelhamentos	356,69
Exponenciações em $\mathbb{G}_T$	295
Multiplicações em $\mathbb{G}_T$	196,8
Multiplicações em $\mathbb{G}$	55,5
Adições em $\mathbb{G}$	35,2

Tabela 7.1: *Tempo de execução para as operações do protocolo proposto*

Para medição dos tempos de execução utilizamos a função `System.currentTimeMillis();` do Java, pois calculam o tempo de execução com grande precisão (em milissegundos).

Durante todo o experimento, o sistema manteve-se isolado de conexões sem fio e da internet, tendo apenas os processos internos como concorrentes. Com intuito de obter uma medida precisa do tempo de execução dos algoritmos, utilizamos a ferramenta de "*profile*" do NetBeans, pois essa ferramenta exclui o tempo gasto pelo escalonador de processos e por outros processos concorrentes do sistema operacional.

Desse modo, os tempos foram obtidos a partir da média de 500 rodadas para cada operação.

As funções hash utilizadas possuem tamanho de 256 bits, para maior compatibilidade com o algoritmo simétrico AES de 128 bits (Terada, 2008), além de ser uma margem confortável para o nível de segurança do protocolo proposto e seguindo a Tabela 7.2 do NIST que apresenta o nível de segurança para um algoritmo assimétrico com equivalencia para um simétrico (AES) que será o mesmo adotado para a função de hash.

Nível de segurança	Simétrico	Assimétrico (RSA)	ECC	Hash
Proteção a curto prazo para pequenas organizações. Não deveriam ser usadas para confidencialidade em novos sistemas	64	816	128	128
Proteção a curto prazo p/ organizações médias a médio prazo para pequenas organizações	72	1008	144	144
Chaves do 3DES restritas a 2^{40} plaintext/ ciphertexts. Proteção até 2012	80	1248	160	160
Legado do nível padrão 2-key 3DES restrito a 10^6 plaintext/ciphertexts. Proteção até 2020	96	1776	192	192
Proteção a médio prazo do 3-key 3DES, até 2030	112	2432	224	224
Proteção a longo prazo. Recomendação genérica independente da aplicação. Proteção até 2040	128	3248	256	256

Tabela 7.2: *Tamanhos dos parâmetros de cada tipo de sistema em função do nível de segurança*

Antes do dispositivo móvel funcionar, necessitamos inicializá-lo – operação descrita na Seção 5.5 (pág.55) – para que o KGC conheça as chaves parciais de cada usuário por um meio seguro. Essa etapa será utilizada como um meio burocrático de cadastro do dispositivo e do usuário. Assegurando

que apenas um usuário e o dispositivo móvel que estiver registrado (certificado) no KGC poderá obter informações da base de dados do Borboleta.

Além disso, a grande quantidade de dados utilizados no cálculo de multiplicação do segredo s por um ponto P da curva requer um poder computacional que sobrecarregaria o celular, desse modo utilizaremos a vantagem que a operação só será realizada apenas uma vez e faremos os cálculos com auxílio de um computador portátil.

Os resultados obtidos nos testes do protocolo proposto são apresentados na tabela a seguir:

Etapas	Tempo de execução (ms) no PC	Tempo de execução (ms) no PDA
Inicialização	98,9	193,7
Chaves do usuário	56,4	63,2
Troca de mensagem	64,5	66,4
Cálculo da chave de sessão	740,9	745,5
Tempo total do protocolo inteiro	844,2	847,1

Tabela 7.3: *Tempo de execução para o protocolo proposto*

Os tempos foram obtidos a partir da média de 500 rodadas do protocolo.

Após a realização do acordo de chave, o sistema precisa efetuar a transmissão segura dos prontuários médicos, entre o PDA e o servidor de banco de dados.

Nesta etapa, os experimentos consistem na execução 500 rodadas do algoritmo simétrico AES para cifragem dos dados do prontuário médico, utilizando a chave de sessão obtida no acordo de chave.

Em cada rodada, foram utilizados 10 arquivos gerados aleatoriamente de 8Mb cada[1], simulando transmissão dos prontuários médicos.

Os resultados obtidos nesses testes são apresentados na tabela a seguir:

Etapas	Tempo de execução (ms) no PC	Tempo de execução (ms) no PDA
Cifragem	300,4	383,9
Decifragem	413,5	478,8

Tabela 7.4: *Tempo de execução para transmissão dos dados*

[1] obtidos em http://www.random.org/files/ (visitado em 1/8/2011)

Capítulo 8

Conclusões

Neste capítulo, vamos apresentar a interpretação dos resultados para concluir a dissertação. E as listas de sugestões para pesquisas futuras.

8.1 Considerações Finais

Este trabalho, tinha como o objetivo principal a construção de um mecanismo provedor de segurança e integridade das informações sigilosas dos dados dos prontuários médicos trafegados no sistema Borboleta.

O protocolo provê autenticação de ambas as partes envolvidas, para isso utilizamos um acordo de chaves baseado em criptografia sem certificados, demonstrado seguro para o problema Gap bilinear Diffie-Hellman computacional (Lippold *et al.*, 2009).

Apesar de parecer pouco eficiente, o nível de segurança alcançado pode ser relaxado para obter um desempenho mais eficiente.

O sistema consiste na implementação de um curva elíptica ordinária BN, com nível de segurança de 128 bits, utilizando emparelhamento Ate. Para gerar ferramentas para implementação do protocolo proposto.

8.2 Sugestões para Pesquisas Futuras

As sugestões para trabalhos futuros são variadas e listadas a seguir para área de pesquisa envolvendo curvas elípticas :

Curvas Edwards desenvolvimento de uma aplicação para verificar o desempenho das coordenadas Edwards, tanto para operações de curvas como para emparelhamentos (Taverne *et al.*, 2011);

Curvas hiperelípticas supersingulares Aranha *et al.* (2010) apresentam uma implementação de curvas hiperelípticas supersingulares com grau de mergulho $k = 12$, isso poderia ser comparado com o desempenho das curvas BN;

Protocolos de criptografia sem certificado avaliar os outros tipos de protocolos sem certificados como, cifra, assinatura e cifrassintura, afim de verificar suas seguranças comparado com os protocolos de acordo de chaves. Ou encontrar uma solução mais eficiênte para este nível de segurança.

Hash eficiente para $\mathbb{G}_2$ técnicas mais avançadas de hashing como de Icart (2009) não são atualmente aplicáveis para nenhuma curva BN. Encontrar um método seguro de hashing daquele tipo para os grupos $\mathbb{G}_1$ e $\mathbb{G}_2$ ou descrever uma subfamília de curvas BN onde tal método seja possível é de grande importância para muitos protocolos baseados em emparelhamento.

Pesquisas envolvendo outros problemas extrapolando o DLP em curvas elípticas, existem os problemas sobre reticulados envolvendo o problema de encontrar um vetor mais curto, que é base neste espaço. Acredita-se que seja um problema mais difícil e por outros fatores seria um problema que superaria a fatoração em computadores pós-quanticos.

Apêndice A

Conceitos Básicos e Propriedades

A.1 Grupo

Um *grupo* $(\mathbb{G}, \circ)$ é um conjunto $\mathbb{G}$ com uma operação binária $\circ$ definida sobre $\mathbb{G}$, tal que:

1. A operação $\circ$ é associativa, i.e.,

$$\forall a, b, c \in \mathbb{G} : a \circ (b \circ c) = (a \circ b) \circ c;$$

2. Existe um elemento $n \in \mathbb{G}$, chamado identidade ou elemento neutro, tal que

$$\forall a \in \mathbb{G} : a \circ n = n \circ a = a;$$

3. Para cada elemento $a \in \mathbb{G}$ existe um elemento $\overline{a} \in \mathbb{G}$, o inverso de a, tal que

$$a \circ \overline{a} = \overline{a} \circ a = e.$$

Dizemos que $\mathbb{G}$ é *comutativo* (ou *abeliano*) se $\forall a, b \in \mathbb{G} : a \circ b = b \circ a$. Se a operação $\circ$ for uma multiplicação $(*)$, o grupo é chamado *grupo multiplicativo*, a identidade é representada por 1 e o inverso de a é representado por a^{-1}. Se a operação $\circ$ for uma soma $(+)$, o grupo é chamado *grupo aditivo*, a identidade é representada por 0 e o inverso de a é representado por $-a$.

Um exemplo de grupo aditivo é o conjunto $\mathbb{Z}$ dos inteiros com a operação de soma usual. Outro exemplo é o conjunto $\mathbb{Z}_n$ dos inteiros $\mod n$, com a soma $\mod n$.

Para as definições a seguir considere que estamos trabalhando com grupos aditivos.

Se o número de elementos de $\mathbb{G}$ for finito, tal número é chamado de *ordem* de $\mathbb{G}$. Se existe um elemento $g \in G$ tal que qualquer elemento $x \in G$ possa ser escrito como um múltiplo escalar de G, i.e., $x = rg$ para algum inteiro r, então dizemos que $\mathbb{G}$ é *cíclico* e $\langle g \rangle$ é o *gerador* de $\mathbb{G}$.

A *ordem de um elemento* $x \in \mathbb{G}$ é o menor inteiro r tal que $rx = 0$. Se não existir esse elemento, dizemos que a ordem de $x \in \mathbb{G}$ é infinita.

Se $\mathbb{G}$ tem ordem prima, $\mathbb{G}$ é cíclico e qualquer $x \in G$ é um gerador de G de ordem r.

Um *subgrupo* de um grupo G é um subconjunto H de G que também é um grupo para a mesma operação. Sejam $(G, *)$ um grupo e H um subconjunto de G. Dizemos que H é um subgrupo de G se verifica:

- H é não vazio;
- H é fechado para a operação de G, isto é, dado dois elementos a, b de H, o resultado da operação $a * b$ também está em H;
- e H é um grupo.

A.1.1 Homomorfismo

Uma aplicação $f : \mathbb{G} \to \mathbb{H}$ de um grupo $\mathbb{G}$ em um grupo $\mathbb{H}$ é chamado um *homomorfismo* se f preserva as operações de $\mathbb{G}$. Isto é, se $*$ e $\bullet$ são operações $\mathbb{G}$ e $\mathbb{H}$, repectivamente, então f preserva as operações de $\mathbb{G}$ se para todos $a, b \in \mathbb{G}$ termos $f(a * b) = f(a) \bullet f(b)$.

Seja $f : \mathbb{G} \to \mathbb{H}$ um homomorfismo, dizemos que:

- é um *monomorfismo*, se for injectivo;
- é um *epimorfismo*, se for sobrejectivo;
- é um *isomorfismo*, se for simultanemente um monomorfismo e um epimorfismo, ou seja, se for uma bijecção;
- é um *endomorfismo*, se $\mathbb{G} = \mathbb{H}$;
- é um *automorfismo*, se for simultaneamente um endomorfismo e um isomorfismo.

Se f for um isomorfismo, então tem uma inversa (pois é uma bijecção). A função f^{-1} é também um homomorfismo de grupos e, portanto, um isomorfismo.

A.1.2 Relação de equivalência

Uma relação de equivalência é uma relação ao mesmo tempo reflexiva, simétrica e transitiva.

Seja $\mathbb{G}$ um grupo. Uma relação de equivalência em $\mathbb{G}$ induz a *partição* de $\mathbb{G}$ em subconjuntos não-vazios mutuamente disjuntos. Esses conjuntos são denominados *classes de equivalência.*

A.2 $\mathbb{Z}$, $\mathbb{Z}_n$ e $\mathbb{Z}_n^*$

$\mathbb{Z}$ é o conjunto de todos os inteiros com operação de soma e multiplicação usuais.

$\mathbb{Z}_n$ é o conjunto de todos os inteiros $\mod n$ com operações de soma e multiplicação $\mod n$. Exemplo: $\mathbb{Z}_{11} = \{0, 1, \ldots, 10\}$.

$\mathbb{Z}_n^*$ é o conjunto de todos os inteiros $\mod n$ relativamente primos a n. Exemplo: $\mathbb{Z}_{10}^* = \{1, 3, 7, 9\}$.

$\mathbb{Z}_n$ é um anel. Se n é primo, então $\mathbb{Z}_{10}^*$ é um corpo.

A.3 Corpo

Um corpo $\mathcal{K}$ é um anel comutativo no qual todos os elementos não nulos possuem inverso multiplicativo, ou seja:

$$\forall a \in \mathcal{K}, a \neq 0 \Rightarrow \exists b \in \mathcal{K} | ab = 1.$$

A característica de um corpo é zero se $\overbrace{1+1+1+\ldots+1}^{m \text{ vezes}}$ é diferente de zero para qualquer $m > 1$. Caso contrário, a característica do corpo é o menor m para o qual esta soma é zero.

A.3.1 Corpo finito

Se o número de elementos do corpo é finito, ele é chamado de corpo finito (*finite field*) ou **Corpo Finito de Galois**.

A ordem de um corpo finito é o número de elementos no corpo. Existe um corpo finito $\mathbb{F}_q$ de ordem q se e somente se q é uma potência prima, isto é, $q = p^m$, onde p é um número primo denominado característica de $\mathbb{F}_q$, e m é um inteiro positivo. Se $p = 2$ então $\mathbb{F}$ é conhecido como corpo binário, ou corpo de característica 2. Se $p > 2$ e $m = 1$ então $\mathbb{F}_q$ é conhecido como um corpo primo ou corpo de característica prima p. Se $p \geq 2$ e $m \geq 2$, então $\mathbb{F}_q$ é um corpo estendido de característica p.

Propriedade A.1. Se $\mathbb{F}_{p^k}$ é um corpo finito então $\mathbb{F}^*_{p^k}$ é um grupo cíclico de ordem $q^k - 1$. Em particular, se $a \in \mathbb{F}^*_{p^k}$, então $a^{p^k-1} = 1$.

A.3.2 Corpo primo

Seja $p > 2$ e $m = 1$. Os inteiros módulo p, consistindo dos inteiros $\{0, 1, 2, \ldots, p-1\}$ com adição e subtração módulo p, é um corpo finito de ordem p. Denominamos este corpo por $\mathbb{F}_p$.

Exemplo A.1 (corpo $\mathbb{F}_{29}$)**.** Os elementos de $\mathbb{F}_{29}$ são $\{0, 1, 2, \ldots, 28\}$. Alguns exemplos de operações aritméticas em $\mathbb{F}_{29}$ são:
Adição: $17 + 20 = 8$ desde que $37 \mod 29 = 8$
Subtração: $17 - 20 = 26$ desde que $-3 \mod 29 = 26$.
Multiplicação: $17 \cdot 20 = 21$ desde $340 \mod 29 = 21$
Inversão: $17^{-1} = 12$ desde $17 \cdot 12 \mod 29 = 1$.

A.3.3 Corpo binário

Corpos finitos de ordem 2^m são chamados de corpos binários, corpos de característica 2. Uma maneira de construir $\mathbb{F}_{2^m}$ é usar a *representação na base polinomial*. Nesta representação os elementos de $\mathbb{F}_{2^m}$ são polinômios binários (polinômios cujos coeficientes estão no corpo $\mathbb{F}_2 = \{0, 1\}$) de grau $\leq m - 1$:

$$\mathbb{F}_{2^m} = \{a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_2x^2 + a_1x^1 + a_0 : a_i \in \{0, 1\}\}$$

Um polinômio binário irredutível $f(x)$ de grau m é escolhido (tal polinômio existe para m e pode ser eficientemente computado). A irredutividade de $f(x)$ significa que $f(x)$ não pode ser fatorado como um produto de polinômios binários de grau menor do que m. A adição de elementos do

corpo é a adição usual de polinômios, com os coeficientes reduzidos a módulo 2. A multiplicação de elementos do corpo é feita pela redução polinomial $f(x)$.

Exemplo A.2 ($\mathbb{F}_{2^4}$)**.** Neste caso o polinômio irredutível é $f(x) = x^4 + x + 1$. Vamos representar o polinômio $b_3x^3 + b_2x^2 + b_1x^1 + b_0$ pelo vetor (b_3, b_2, b_1, b_0).

Por exemplo:

$$(1011) \oplus (1001) = (0010)$$

$$(1101) \otimes (1001) = (1111),$$

pois $(x^3 + x^2 + 1)(x^3 + 1) = x^6 + x^5 + x^2 + 1 = (x^3 + x^2 + x + 1) \mod f(x)$.

$\mathbb{F}_{2^4} - \{0\}$ forma um grupo multiplicativo de ordem 15: $g = (0010)$ é o gerador deste grupo:

$$\begin{array}{lllll} g^1 = (0010), & g^2 = (0100), & g^3 = (1000), & g^4 = (0011), & g^5 = (0110), \\ g^6 = (1100), & g^7 = (1011), & g^8 = (0101), & g^9 = (1010), & g^{10} = (0111), \\ g^{11} = (1110), & g^{12} = (1111), & g^{13} = (1101), & g^{14} = (1001), & g^{15} = (0001). \end{array}$$

A.3.4 Representação de corpos finitos $\mathbb{F}_{p^2}$

Quando o polinômio $x^2 + 1$ é irredutível no anel $\mathbb{F}_p[x]$?

Se ele for redutível, então fatores como: $x^2 + 1 = (a + \alpha)(x + \beta)$ para algum $\alpha, \beta \in \mathbb{F}_p$.

Comparando os coeficientes, vemos que $\alpha + \beta = 0$ e $\alpha\beta = 1$; consequentemente $\alpha^2 = \alpha.(-\beta) = -\alpha\beta = -1$.

Em outras palavras, o corpo $\mathbb{F}_p$ tem um elemento cujo quadrado é -1. Por outro lado, se $\alpha \in \mathbb{F}_p$ satisfaz $\alpha^2 = -1$, então $x^2 + 1 = (x - \alpha)(x + \alpha)$ fatores de $\mathbb{F}_p[x]$.

Isso prova que $x^2 + 1$ é irredutível em $\mathbb{F}_p[x]$ se e somente se -1 não é um quadrado em $\mathbb{F}_p$.

Reciprocidade Quadrática nos diz que x^2+1 é irredutível em $\mathbb{F}_p[x]$ se e somente se $p \equiv 3 (\text{mod } 4)$.

Seja p um primo que satisfaça $p \equiv 3 (\text{mod } 4)$. Então, o corpo quociente $\mathbb{F}_p[x]/(x^2 + 1)$ é um corpo que contém p^2 elementos. Ele contém um elemento $\hat{x}$ que é a raiz quadrada de -1. Assim, podemos ver $\mathbb{F}_p[x]/(x^2+1)$ como uma espécie de análogo de números complexos e podemos escrever os seus elementos na forma

$$a + bi \text{ com } a, b \in \mathbb{F}_p,$$

onde i é simplesmente um símbolo com a propriedade que $i^2 = -1$.

Adição, subtração, multiplicação e divisão são executadas apenas como em números complexos, com o entendimento que, em vez de números reais como coeficientes, estamos usando inteiros modulo p. Assim, por exemplo, a divisão é feita pelo usual truque "racionalizar o denominador",

A.3.5 Simetria rotacional

Formalmente, a simetria rotacional é a simetria com respeito a algumas ou todas rotações em espaço euclidiano m-dimensional.

Rotações são isometrias diretas, ou seja, isometrias preservando a orientação.

Portanto, um grupo de simetria de simetria rotacional é um subgrupo de $E^+(m)$.

Simetria com respeito a todas as rotações sobre todos os pontos implica a simetria de translação com respeito a todas as translações, por isso o espaço é homogêneo, e o grupo de simetria é o $E(m)$ todo.

Para simetria com respeito às rotações sobre um ponto, pode-se considerar este ponto como origem.

Estas rotações formam o grupo especial ortogonal $SO(m)$, o grupo de matrizes ortogonais $m \times m$ com determinante 1.

Para $m = 3$ este é o grupo de rotação.

Em outro sentido da palavra, o grupo de rotação de um objeto é o grupo de simetria no interior de $E^+(n)$, o grupo de isometrias diretas; em outras palavras, a interseção do grupo de simetria completo e o grupo de isometrias diretas.

Para objetos quirais[1] é o mesmo que o grupo de simetria completo.

A.3.6 Simetria rotacional de ordem n

A simetria rotacional de ordem n, também chamada de n-fold simetria rotacional, ou simetria rotacional discreta de ordem n, com relação a um determinado ponto (no $\mathbb{R}^2$) ou eixo (no $\mathbb{R}^3$) significa que a rotação por um ângulo de $360°/n$ (180°, 120°, 90°, 72°, 60°, 513/7°, entre outros) não altera o objeto.

Note que simetria "1-vez" não é simetria, e "2-vezes" é a simetria mais simples, por isso não quer dizer "a mais básica".

A notação de simetria n-vezes é $\mathbb{C}_n$, ou simplesmente "n".

Para cada ponto ou eixo de simetria o tipo de grupo abstrato é grupo cíclico $\mathbb{Z}_n$ de ordem n.

A.3.7 Raizes n-ésimas da unidade

Definição A.1. *Circunferência unitária* - Dado $G = S^1$, o conjunto dos números complexos de módulo 1, com a operação que é o produto usual de números complexos.

Definição A.2. As *raízes n-ésimas da unidade*, ou números de *Moivre*, são todos os números complexos que resultam 1 quando são elevados a uma dada potência n. Estas raízes estão localizadas no círculo unitário.

O problema da determinação das raízes da unidade consiste na obtenção de todos os números complexos que satisfaçam uma equação polinomial da forma

$$x^n - 1 = 0$$

onde n é um número natural. É evidente da equação que a unidade é uma solução. Vemos também que qualquer equação da forma

$$x^n - a^n = 0$$

[1] que conduz ao termo *quiralidade*, é um termo usado em Química, para definir objetos não sobreponíveis à sua própria imagem no espelho.

pode-se reduzir à anterior por intermédio da substituição $y = x/a$. Se b for um número tal que $b^n = a$, dizemos que b é uma raiz de índice n de a. Pelo teorema fundamental da álgebra, sabemos que existem n raízes de a, contando com possíveis multiplicidades.

Suponhamos que a é um número real não negativo. Então, existe um número real não negativo b tal que $b^n = a$. Com efeito, consideramos a função $f(x) = x^n$ e verificamos que é possível aplicar o teorema de Bolzano ao intervalo compreendido entre os números $1/a$ e a. O número b nessas condições denota-se por $\sqrt{a}^x$ e lê-se raiz de índice n de a.

Consideremos de novo a equação $x^n - 1 = 0$. Se α é uma raiz da equação dada então também o é α^p pois $(\alpha^p)^n = \alpha^{pn} = (\alpha^n)^p = 1^p = 1$. Daqui resulta imediatamente que se α é uma raiz de $x^n - 1 = 0$ então também é raiz de $x^{np} - 1 = 0$. Deste modo é suficiente estudar a factorização no caso em que n é um número primo.

A.3.8 Operações básicas com números complexos

Dados os números complexos $z = a + bi$ e $w = c + di$, podemos definir duas operações fundamentais, adição e produto, agindo sobre eles da seguinte forma:

$$z + w = (a + bi) + (c + di) = (a + c) + (b + d)i$$

$$z.w = (a + bi).(c + di) = (ac - bd) + (ad + bc)i$$

Observação: Tais operações lembram as operações com expressões polinomiais, pois a adição é realizada de uma forma semelhante, isto é: $(a+bx)+(c+dx) = (a+c)+(b+d)x$ e a multiplicação $(a + bx).(c + dx)$, é realizada através de um algoritmo que aparece na forma:

por exemplo,

$$\begin{array}{cccccc} a & + & bx & & \\ c & + & dx & & \times \\ \hline ac & + & bcx & & \\ & & adx & + & bdx^2 \\ \hline ac & + & (bc+ad)x & + & bdx^2 \end{array}$$

de forma que devemos substituir x^2 por -1.

A divisão é feita pelo uso do truque "racionalizar o denominador",

$$\frac{a + bi}{c + di} = \frac{a + bi}{c + di} \cdot \frac{c - di}{c - di} = \frac{(ac + bd) + (bc - ad)i}{c^2 + d^2}$$

A fim de construir um corpo com elementos p^d, precisamos encontrar um polinômio irredutível de grau d em $\mathbb{F}_p[x]$.

Está provado nos textos mais avançados que há sempre um polinômio tal, e na verdade geralmente muitos desses polinômios.

Além disso, em certo sentido abstrato, não importa que nós escolhemos polinômio irredutível: nós sempre temos o mesmo domínio. No entanto, em um sentido prático que faz uma diferença, porque os cálculos práticos $\mathbb{F}_p[x]/(\mathbf{m})$ são mais eficientes se $\mathbf{m}$ não tem muitos coeficientes diferentes de zero.

A.4 Curiosidades

Definição A.3. Arnold Schönhage define *quolinômio*[2] como o quociente p/q de dois polinômios (possivelmente multivariados), por oposição ao termo "função racional", que pode ser confundido com $f \in \mathbb{Q}[X]$.

[2]tradução do inglês *quolynomial*

Apêndice B

Divisores

Um *divisor* é um caminho para caracterizar uma função f baseada somente em zeros, onde $f(x) = 0$, e pólos, onde $f(x) = \pm\infty$, como quando dividimos por zero. Nós dizemos que uma função $f(x)$ tem um pólo no infinito se $f(1/x)$ tem um pólo em $x = 0$, de modo que um polinômio de grau n tem um pólo de grau n no infinito. Analogamente, nós dizemos que uma função $f(x)$ tem um zero no infinito se $f(1/x)$ tem um zero em $x = 0$. Por exemplo, a função

$$f(x) = \frac{(x-1)^2}{(x+2)^3} = (x-1)^2(x+2)^{-3}$$

tem um zero de ordem 2 em $x = 1$, um zero de ordem 1 no infinito, e um pólo de ordem 3 em $x = -2$. Porque um divisor caracteriza uma função baseda em zeros e pólos, duas funções que se diferem por uma constante terá o mesmo divisor.

B.1 Uma introdução intuitiva para divisores

Mantendo em contato com os zeros e pólos de uma função racional f em que nós denotamos um divisor, que nós escrevemos como $div(f)$. Nós escrevemos tal divisor como a soma de pontos onde f tem um zero ou peso do pólo pelas multiplicidades de zeros e pólos, com a convenção que zeros têm pesos positivos conforme suas multiplicidades e pólos têm pesos negativos conforme suas multiplicidades. No exemplo acima, nós escrevemos $div(f) = 2(1) + (\infty) - 3(-2)$, para indicar que f tem um zero de ordem 2 em $x = 1$, um zero ou ordem 1 no infinito, e um pólo de ordem 3 em $x = -2$. Em geral, se nós podemos escrever

$$f(x) = \prod_i (x - x_i)^{a_i}$$

então nós escrevemos

$$div(f) = \sum_i a_i(x_i)$$

A notação para divisores pode ser um pouco complicada, e nós vamos precisar ser capazes de distinguir o contexto que estamos lidando com divisores em vez de números, de modo que não somos tentados a tratar como divisores de números, tentando simplificar expressões como $2(1) - 3(-2)$ para obter um número ao invés de um divisor.

Note que ao multiplicar as funções racionais corresponde a adicionar seus divisores e divisão de funções racionais corresponde a subtração de seus divisores. Então, se temos $f(x)$, conforme definido acima e

$$g(x) = \frac{(x+2)^3}{(x+1)^4}$$

então

$$\begin{aligned} f(x)g(x) &= \frac{(x-1)^2}{(x+2)^3}\frac{(x+2)^3}{(x+1)^4} \\ &= \frac{(x-1)^2}{(x+1)^4} \end{aligned}$$

que corresponde à adição dos divisores:

$$\begin{aligned} div(fg) &= div(f) + div(g) \\ &= 2(1) + (\infty) - 3(-2) + 3(-2) + (\infty) - 4(-1) \\ &= 2(1) + 2(\infty) - 4(-1) \end{aligned}$$

Podemos formalizar esta descrição informal dos divisores com as seguintes definições.

Definição B.1. Uma *soma formal* de um conjunto S é uma série $\{s_0, s_1, s_2, \ldots\}$ de elementos de S. Uma soma formal é frequentemente escritos utilizando um espaço reservado, com o entendimento de que o espaço reservado não está a ser avaliada.

Exemplo B.1.

(i) Uma série de potências é uma soma formal que costumamos escrever como $a_0 + a_1x + a_2x^2 + \ldots$, onde cada $a_i \in S$ para um conjunto S. Nós escrevemos uma série de potência com o entendimento de que o espaço reservado x não está a ser avaliado, e nós também podemos escrever a mesma série de potência como $\{a_0, a_1, a_2, \ldots\}$.

(ii) Se $P = \{P_1, P_2, \ldots, P_n\}$ é um conjunto de pontos sobre uma curva elíptica, então $D = a_1(P_1) + a_2(P_2) + \ldots + a_n(P_n)$ é uma soma formal dos elementos de P. Neste caso, entendemos que em D pontos do conjunto P são apenas espaços reservados, como a variável x em uma série de potências, e não estão a ser avaliados.

Definição B.2. Seja E uma curva elíptica. Um *divisor* de E é uma soma formal da seguinte forma

$$D = \sum_{P \in S} n_P(P)$$

onde cada n_P é um inteiro e todos, exceto esta quantidade n_P finita, são zero.

Exemplo B.2. Para os pontos P_1 e P_2 sobre uma curva elíptica, $D = (P_1) + 2(P_2) - 3(\mathcal{O})$ é um divisor.

Definição B.3. Dizemos que um divisor D é um *divisor principal* , se houver uma função racional f tal que $D = div(f)$. Uma definição equivalente é que um divisor D em uma curva elíptica é principal se podemos escrever

$$D = \sum_i a_i(P_i)$$

onde $\sum a_i = 0$ e $\sum a_i P_i = \mathcal{O}$, com a última soma usando a adição de pontos em uma curva elíptica. Em particular, se P é um ponto de ordem n, então o divisor $n(P) - n(\mathcal{O})$ é um divisor principal.

Exemplo B.3.

(i) Dados P_1, P_2 e P_3 pontos de uma curva elíptica com $P_3 = P_1 + P_2$. Então $D = (P_1) + (P_2) + (-P_3) - 3(\mathcal{O})$ é um divisor principal.

(ii) Seja P um ponto sobre uma curva elíptica de ordem n. Então, $D = n(P) - n(\mathcal{O})$ é um divisor principal.

Definição B.4. Se E é uma curva elíptica e

$$D = \sum_{P \in E} n_P(P)$$

é um divisor, então o *suporte* de D é o conjunto dos pontos P tal que $nP \neq 0$.

Exemplo B.4. Para o divisor $D = (P_1) + (P_2) + (-P_3) - 3(\mathcal{O})$, o apoio do D é o conjunto $\{P_1, P_2, -P_3, \mathcal{O}\}$.

Definição B.5. Deixe D_1 e D_2 ser divisores. Então dizemos que D_1 e D_2 tem *suporte disjunto* se a intersecção do suporte de D_1 e o suporte de D_2 é o conjunto vazio, ou $D_1 \cap D_2 = \emptyset$.

Exemplo B.5.

(i) Os divisores $D_1 = (P_1) - (\mathcal{O})$ e $D_2 = (P_1 + R) - (R)$ tem suporte disjunto, enquanto $\{P_1, \mathcal{O}\} \cap \{P_1 + R, R\} = \emptyset$.

(ii) Os divisores $D_1 = (P) - (\mathcal{O})$ e $D_2 = (Q) - (\mathcal{O})$ não têm suporte disjunto.

Podemos pensar nos divisores como preservador do trajeto de onde o gráfico de uma curva elíptica E intercepta o gráfico de uma função $f(x)$, ou quando $E = f(x)$, para que eles mantenham os zeros e pólos de $E = f(x)$.

Em particular, temos um zero quando $E = f(x)$, ou quando a função $f(x)$ cruza a curva elíptica E e nós temos um pólo quando $f(x)$ tem um pólo.

As funções u e v que aparecem na Figura B.1 são muito importantes na execução de operações de divisores, e no seguinte, u representará sempre uma linha através de dois pontos $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$ em uma curva elíptica e v representará sempre uma linha vertical que passa por $P_3 = (x_3, y_3)$, onde $P_3 = P_1 + P_2$.

Suponha que nós não temos o caso de $P_1 + P_2 = \mathcal{O}$ e nem $P_1 = \mathcal{O}$ nem $P_2 = \mathcal{O}$. Então, podemos escrever a forma ponto-inclinação de uma linha através de (x_1, y_1) como

$$y - y_1 = m(x - x_1)$$

ou

$$y - y_1 = -mx + mx_1 = 0$$

que nos dá de maneira explícita para encontrar a linha de u. Da mesma forma, a linha v é dada por

$$x - x_3 = 0$$

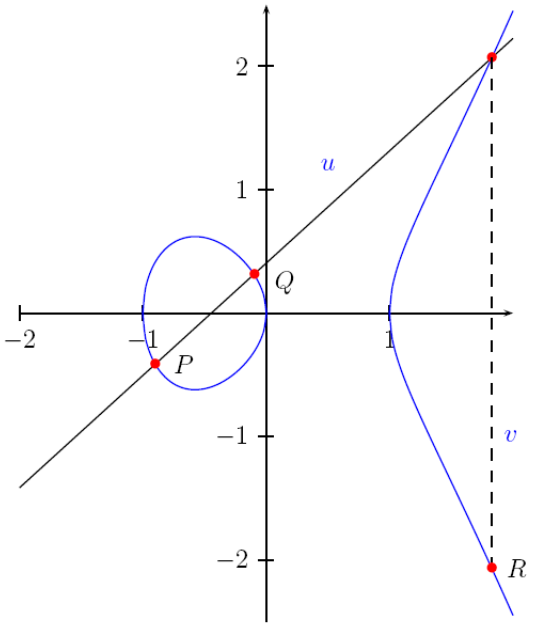


Figura B.1: *Ilustração das linhas u e v na adição de pontos em uma curva elíptica.*

Se um dos dois pontos é $\mathcal{O}$, então u é a linha vertical que passa pelo ponto que não é $\mathcal{O}$, e se o ponto $(x_3, y_3) = \mathcal{O}$ então v é a linha vertical $x = 0$. Estas formas de linhas (x_1, y_1) e (x_1, y_1) são mostrados na Figura B.2. Os casos em que seja $P_1 = \mathcal{O}, P_2 = \mathcal{O}$, ou $P_1 = P_2$ são mostrados nos Algoritmos 3, 4 e 5.

Os pontos particulares que usamos para definir as linhas u e v devem ser claros a partir do contexto, por isso, geralmente omite-se os pontos para manter a notação mais simples.

Se temos de esclarecer que pontos estão sendo usados, vamos escrever u_{P_1,P_2} e v_{P_3} para indicar a linha através de P_1 e P_2 ou a linha vertical através de P_3, respectivamente. Com esta notação, u e v tem os seguintes divisores:

$$\begin{aligned} div(u) &= (P_1) + (P_2) + (-P_3) - 3(\mathcal{O}) \\ div(v) &= (P_3) + (-P_3) - 2(\mathcal{O}) \end{aligned}$$

onde já representaram os pólos que as linhas u e v têm em $\mathcal{O}$.

Outro fato útil é o que obtemos quando subtraimos o divisor de u do divisor de v:

$$\begin{aligned} div(u) - div(v) &= div(u/v) \\ &= (P_1) + (P_2) + (P_3) - (\mathcal{O}) \end{aligned} \tag{B.1}$$

Se tivermos dois divisores da forma:

$$\begin{aligned} D_1 &= (P_1) - (\mathcal{O}) + div(f_1) \\ D_1 &= (P_2) - (\mathcal{O}) + div(f_2) \end{aligned}$$

podemos somar os dois divisores para obter

$$D_1 + D_2 = (P_1) + (P_2) - 2(\mathcal{O}) + div(f_1 f_2) \tag{B.2}$$

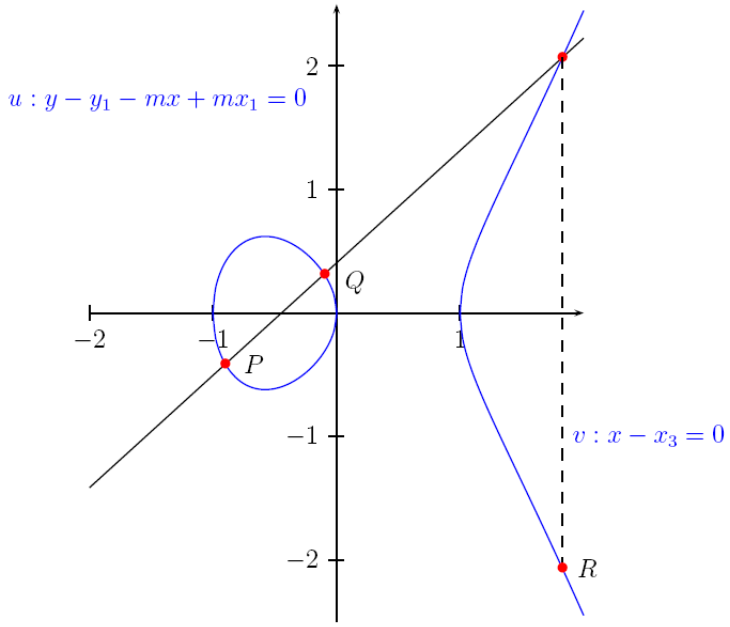


Figura B.2: *Formas das linhas u e v usadas para adicionar divisores de uma curva elíptica.*

Resolvendo para $(P_1) + (P_2)$ em B.1 e substituindo o resultado em B.2 vemos que

$$D_1 + D_2 = (P_3) - (\mathcal{O}) + div(f_1 f_2 u/v) \tag{B.3}$$

Assim, os divisores de linhas de u e v fornece uma maneira de acrescentar dois divisores e manter o resultado na forma de $(P) - (\mathcal{O}) + div(f)$.

Para esclarecer como isso funciona, vamos agora passar por um cálculo da soma de dois divisores, onde a média aritmética é feita sobre a curva $y^2 = x_3 + 1$ sobre $\mathbb{F}_5$.

Particularmente, consideramos o divisor $D = (\hat{P}_2) - (\mathcal{O})$ e veja o que obtemos quando a adicionamos a si mesma. Usando B.3 e o fato de que também podemos escrever o divisor D como $div(1)$ vemos que

$$\begin{aligned} D + D &= (\hat{P}_2) - (\mathcal{O}) + div(1) + (\hat{P}_2) - (\mathcal{O}) + div(1) \\ &= (\hat{P}_1) - (\mathcal{O}) + div(u/v) \end{aligned}$$

Agora, u é a linha tangente à curva elíptica em $\hat{P}_2$, e v é a linha que liga $\hat{P}_2 + \hat{P}_2 = \hat{P}_1$, e $-(\hat{P}_2 + \hat{P}_2) = \hat{P}_2$. Resolvendo para u e v encontramos que temos $y - 4 = 0$ para a linha u, ou $y + 1 = 0$ em $\mathbb{F}_5$. Da mesma forma, temos que $x = 0$ para a linha v. Substituindo estes para u e v, obtemos que

$$D + D = (\hat{P}_1) - (\mathcal{O}) + div\left(\frac{y+1}{x}\right)$$

Se somarmos o divisor D a essa soma mais uma vez, constataremos que estamos apenas a esquerda com o divisor de uma função racional quando os termos do divisor envolvendo pontos na curva anular um ao outro, assim quando obtemos $3D = 3(\hat{P}_2) - 3(\mathcal{O})$ porque $\hat{P}_2$ é um ponto de ordem 3.

No próximo passo, a linha u que passa através de P_1 e P_2 é a linha vertical $x = 0$, pois $x = 0$ é a coordenada comum x que P_1 e P_2 compartilham. Nós definimos o v vertical que passa pelo ponto $\hat{P}_1 + \hat{P}_2 = \mathcal{O}$ sendo 1. Assim, temos

$$\begin{aligned} 3D &= 3(\hat{P}_2) - 3(\mathcal{O}) \\ &= (\hat{P}_2 + \hat{P}_1) - (\mathcal{O}) + div\left(\frac{y+1}{x}\frac{u}{v}\right) \\ &= (\mathcal{O}) - (\mathcal{O}) + div\left(\frac{y+1}{x}\frac{x}{1}\right) \\ &= div(y+1) \end{aligned}$$

Definição B.6. Se D é um divisor da forma

$$D = \sum_i a_i(P_i)$$

então definimos o que significa avaliar uma função racional f em D por

$$f(D) = \prod_i f(P_i)^{a_i}$$

Exemplo B.6.

(i) Se $D = 2(P_1) - 3(P_2)$ então

$$\begin{aligned} f(D) &= f(P_1)^2 f(P_2)^{-3} \\ &= \frac{f(P_1)^2}{f(P_2)^3} \end{aligned}$$

(ii) Se $P = (2,3)$ e $Q = (0,1)$ são pontos de $E/\mathbb{F}_{11}$ e D é o divisor $D = (P) - (Q)$ e f é a função racional $f(x,y) = y + 1$, então

$$f(D) = \frac{3+1}{1+1} = 4 \cdot 2^{-1} = 4 \cdot 6 \equiv$$

Em muitos casos, é possível trocar os papéis de uma função f e um divisor D em expressões como $f(D)$. Isto será formalizado a seguir.

Propriedade B.1. (Reciprocidade de Weil)

Seja f e g funções racionais definidas em algum corpo F. Se $div(f)$ e $div(g)$ tem suporte disjunto, então temos que $f(div(g)) = g(div(f))$.

Exemplo B.7. Suponha que temos duas funções racionais f e g definidas em $\mathbb{F}_{11}$ onde

$$\begin{cases} f(x) = \dfrac{x-2}{x-7} & \text{e} \\ g(x) = \dfrac{x-6}{x-5} \end{cases}$$

de modo que temos

$$\begin{cases} div(f) = (2) - (7) & \text{e} \\ div(g) = (6) - (5) \end{cases}$$

então

$$\begin{cases} f(div(g)) = \dfrac{f(6)}{f(5)} = \dfrac{7}{4} = 7 \cdot 3 = 10(mod\ 11) & \text{e} \\ g(div(f)) = \dfrac{g(2)}{g(7)} = \dfrac{5}{6} = 5 \cdot 2 = 10(mod\ 11) \end{cases}$$

Definição B.7. Divisores D_1 e D_2 são *equivalentes* se eles diferem por um divisor principal, ou seja, $D = D_1 - D_2$ é um divisor principal.

Exemplo B.8.

(i) Se f é uma função racional, os divisores $(P) - (\mathcal{O})$ e $(P) - (\mathcal{O}) + div(f)$ são equivalentes.

(ii) Podemos ver que $(P + R) - (R)$ é equivalente a $(P) - (\mathcal{O})$, usando a linha u que atravessa os pontos P, R e $-(P + R)$ e a linha v que passa pelos pontos $-(P + R)$ e $P + R$. Então temos que

$$\begin{aligned} div(u) &= (P) + (R) + (-(P + R)) - 3(\mathcal{O}) \\ div(v) &= (-(P + R)) + (P + R) - 2(\mathcal{O}) \end{aligned}$$

de modo que

$$(P) - (\mathcal{O}) = (P + R) - (R) + div(u/v)$$

Assim, a diferença entre $(P + R) - (R)$ e $(P) - (\mathcal{O})$ é um divisor principal, já que é o divisor da função racional u/v, e $(P + R) - (R)$ é equivalente a $(P) - (\mathcal{O})$.

Algoritmo 3: Equação da reta Secante que passa pelos pontos P e Q

Entrada: Pontos $P(x_p, y_p), Q(x_q, y_q) \in E[n]$.
Saída: Reta $ax + by + c = 0$ onde $a, b, c \in F_p^2$.

```
1  início
2      a ← 0; b ← 0; c ← 0;
3      se P = ∞ e Q = ∞ então
4          c ← 1;
5      senão
6          x_r ← x_q − x_p;
7          y_r ← y_q − y_p;
8          a ← −y_r;
9          b ← x_r;
10         c ← −(a x_r + b y_r);
11     devolve ax + by + c;
12 fim
```

Algoritmo 4: Equação da reta Tangente ao ponto P

Entrada: Ponto $P(x_p, y_p) \in E[n]$.

Saída: Reta $ax + by + c = 0$ onde $a, b, c \in F_p^2$.

```
1  início
2      a ← 0; b ← 0; c ← 0;
3      se P = ∞ então
4          c ← 1;
5      senão se y_p = 0 então
6          a ← 1;
7          c ← −x_p;
8      senão
9          m ← (3x_p^2 + a_4)/(2y_p);
10         a ← −m;
11         b ← 1;
12         c ← (−x_p^3 + a_4 x_p + 2a_6)/(−2y_p);
13     devolve ax + by + c;
14 fim
```

Algoritmo 5: Equação da reta Vertical no ponto P

Entrada: Ponto $P(x_p, y_p) \in E[n]$.

Saída: Reta $ax + by + c = 0$ onde $a, b, c \in F_p^2$.

```
1  início
2      a ← 0; b ← 0; c ← 0;
3      se P = ∞ então
4          c ← 1;
5      senão
6          a ← 1;
7          c ← −x_p;
8      devolve ax + by + c;
9  fim
```

Referências

1363-2000(2000) IEEE Standard Number 1363-2000. Standard specifications for public-key cryptography. Relatório técnico, IEEE Computer Society. URL http://ieeexplore.ieee.org/servlet/opac?punumber=7168. Citado na pág. 63

Al-Riyami e Paterson(2003) Sattam S. Al-Riyami e Kenneth G. Paterson. Certificateless public key cryptography. Em *ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*. Springer. ISBN 3-540-20592-6. Cryptology ePrint Archive, Report 2003/126, http://eprint.iacr.org/2003/126. Citado na pág. 8, 13, 28

Aranha *et al.***(2010)** Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys e Julio López. Faster explicit formulas for computing pairings over ordinary curves. Cryptology ePrint Archive, Report 2010/526, 2010. http://eprint.iacr.org/2010/526. Citado na pág. 75

Atkin e Morain(1993) A. O. L. Atkin e F. Morain. Elliptic curves and primality proving. *Math. Comp*, 61:29–68. Citado na pág. 63

Barreto *et al.***(2007)** Paulo Barreto, Steven Galbraith, Colm hÉigeartaigh e Michael Scott. Efficient pairing computation on supersingular abelian varieties. *Designs, Codes and Cryptography*, 42: 239–271. ISSN 0925-1022. URL http://dx.doi.org/10.1007/s10623-006-9033-6. 10.1007/s10623-006-9033-6. Citado na pág. 41

Barreto *et al.***(2002)** Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn e Michael Scott. Efficient algorithms for pairing-based cryptosystems. Em *CRYPTO '02: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, páginas 354–368, London, UK. Springer-Verlag. ISBN 3-540-44050-X. Citado na pág. 64

Barreto *et al.***(2005)** Paulo S. L. M. Barreto, Michael Naehrig, Escola Politécnica e Lehrstuhl Für Theoretische Informationstechnik. Pairing-friendly elliptic curves of prime order. Em *Proceedings of SAC 2005, volume 3897 of LNCS*, páginas 319–331. Springer-Verlag. Citado na pág. 45, 46, 64, 65

Bellare e Rogaway(1993) Mihir Bellare e Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. Em *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, páginas 62–73, New York, NY, USA. ACM. ISBN 0-89791-629-8. doi: http://doi.acm.org/10.1145/168588.168596. Citado na pág. 9, 10

Bellare *et al.***(2003)** Mihir Bellare, Alexandra Boldyreva e Adriana Palacio. A separation between the random-oracle model and the standard model for a hybrid-encryption problem, 2003. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.4282. Citado na pág. 11

Bentahar *et al.***(2008)** K. Bentahar, P. Farshim, J. Malone-Lee e N.P. Smart. Generic constructions of identity-based and certificateless kems. Cryptology ePrint Archive, Report 2005/058, 2008. ISSN 0933-2790. http://eprint.iacr.org/2005/058. Citado na pág. 47

Blake *et al.***(2005)** I. Blake, G. Seroussi, N. Smart e J. W. S. Cassels. *Advances in Elliptic Curve Cryptography (London Mathematical Society Lecture Note Series)*. Cambridge University Press, New York, NY, USA. ISBN 052160415X. Citado na pág. 24

Boneh e Franklin(2001) Dan Boneh e Matthew K. Franklin. Identity-based encryption from the weil pairing. Em *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, páginas 213–229, London, UK. Springer-Verlag. ISBN 3-540-42456-3. URL http://eprint.iacr.org/2001/090. Citado na pág. 8, 31, 52

Brezing e Weng(2005) Friederike Brezing e Annegret Weng. Elliptic curves suitable for pairing based cryptography. *Designs, Codes and Cryptography*, 37:133–141. ISSN 0925-1022. doi: 10.1007/s10623-004-3808-4. URL http://portal.acm.org/citation.cfm?id=1081027.1081035. Citado na pág. 64

Brier *et al.***(2009)** Eric Brier, Jean-Sebastien Coron, Thomas Icart, David Madore, Hugues Randriam e Mehdi Tibouchi. Efficient indifferentiable hashing into ordinary elliptic curves. Cryptology ePrint Archive, Report 2009/340, 2009. http://eprint.iacr.org/. Citado na pág. 67

Canetti *et al.***(2004)** Ran Canetti, Oded Goldreich e Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/1008731.1008734. Citado na pág. 11

Cash *et al.***(2008)** David Cash, Eike Kiltz e Victor Shoup. The twin diffie-hellman problem and applications. Em *EUROCRYPT'08: Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology*, páginas 127–145, Berlin, Heidelberg. Springer-Verlag. ISBN 3-540-78966-9, 978-3-540-78966-6. Citado na pág. 12, 14

Catalano *et al.***(2005)** Dario Catalano, Ronald Cramer, Ivan Damgard, Giovanni Di Crescenzo, David Pointcheval e Tsuyoshi Takagi. *Contemporary Cryptology (Advanced Courses in Mathematics - CRM Barcelona)*. Birkhauser. ISBN 376437294X. Citado na pág. 1, 5, 10

Chatterjee e Menezes(2009) Sanjit Chatterjee e Alfred Menezes. On cryptographic protocols employing asymmetric pairings – the role of ψ revisited. Cryptology ePrint Archive, Report 2009/480, August 2011 2009. http://eprint.iacr.org/2009/480. Citado na pág. 31

Chatterjee *et al.***(2005)** Sanjit Chatterjee, Palash Sarkar e Rana Barua. Efficient computation of tate pairing in projective coordinate over general characteristic fields. Em Choon-sik Park e Seongtaek Chee, editors, *Information Security and Cryptology - ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, páginas 257–283. Springer Berlin / Heidelberg. URL http://dx.doi.org/10.1007/11496618_13. Citado na pág. 67

Chen *et al.***(2007)** L. Chen, Z. Cheng e N. Smart. Identity-based key agreement protocols from pairings. *International Journal of Information Security*, 6:213–241. ISSN 1615-5262. URL http://dx.doi.org/10.1007/s10207-006-0011-9. 10.1007/s10207-006-0011-9. Citado na pág. 10, 30, 53

Cheng *et al.***(2007)** Zhaohui Cheng, Liqun Chen, Li Ling e Richard Comley. General and efficient certificateless public key encryption constructions. Em *Pairing*, volume 4575 of *Lecture Notes in Computer Science*, páginas 83–107. Springer. Citado na pág. 13, 50

Chung e Hasan(2007) Jaewook Chung e M. Anwar Hasan. Asymmetric squaring formulae. Em *Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, páginas 113–122, Washington, DC, USA. IEEE Computer Society. ISBN 0-7695-2854-6. doi: 10.1109/ARITH.2007.11. URL http://portal.acm.org/citation.cfm?id=1270377.1270455. Citado na pág. 62

Darrel Hankerson(2004) Scott Vanstone Darrel Hankerson, Alfred J. Menezes. *Guide to Elliptic Curve Cryptography*. Springer Professional Computing. Springer. Citado na pág. 25

de Holanda Ferreira e outros(1999) Aurélio Buarque de Holanda Ferreira e outros. *Aurélio Século XXI: O Dicionário da Língua Portuguesa*. Nova Fronteira, 3 edição. Citado na pág. 1

Dent(2008) Alexander W. Dent. A survey of certificateless encryption schemes and security models. *Int. J. Inf. Secur.*, 7(5):349–377. ISSN 1615-5262. doi: http://dx.doi.org/10.1007/s10207-008-0055-0. Cryptology ePrint Archive, Report 2006/211, http://eprint.iacr.org/2006/211. Citado na pág. 9, 48

Devegili *et al.***(2007)** Augusto Jun Devegili, Michael Scott e Ricardo Dahab. Implementing cryptographic pairings over barreto-naehrig curves, in: Pairing-based cryptography pairing 2007. *LNCS*, 4575:197–207. Citado na pág. 66

Diffie e Hellman(1976) Whitfield Diffie e Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654. URL http://citeseer.ist.psu.edu/diffie76new.html. Citado na pág. 7

Dupont e Enge(2002) Régis Dupont e Andreas Enge. Practical non-interactive key distribution based on pairings. Cryptology ePrint Archive, Report 2002/136, 2002. http://eprint.iacr.org/2002/136. Citado na pág. 51, 55

Fiat e Shamir(1987) Amos Fiat e Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. Em Andrew Odlyzko, editor, *Advances in Cryptology - CRYPTO' 86*, volume 263 of *Lecture Notes in Computer Science*, páginas 186–194. Springer Berlin / Heidelberg. URL http://dx.doi.org/10.1007/3-540-47721-7_12. 10.1007/3-540-47721-7_12. Citado na pág. 9

Freeman(2006) David Freeman. Constructing pairing-friendly elliptic curves with embedding degree 10. Em *10th Workshop on Elliptic Curves in Cryptography (ECC 2006*, páginas 452–465. Springer-Verlag. Citado na pág. 64

Galbraith(2001) Steven D. Galbraith. Supersingular curves in cryptography. Em *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '01, páginas 495–513, London, UK. Springer-Verlag. ISBN 3-540-42987-5. URL http://portal.acm.org/citation.cfm?id=647097.717012. Citado na pág. 37

Galbraith *et al.***(2006)** Steven D. Galbraith, Kenneth G. Paterson e Nigel P. Smart. Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165, 2006. http://eprint.iacr.org/2006/165. Citado na pág. 29

Goldreich *et al.***(1986)** Oded Goldreich, Shafi Goldwasser e Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/6490.6503. Citado na pág. 11

Goya(2006) D. H. Goya. Proposta de esquemas de criptografia e de assinatura sob modelo de criptografia de chave pública sem certificado. Dissertação de Mestrado, Universidade de São Paulo. http://www.ime.usp.br/~dhgoya. Citado na pág. 10

Goya *et al.***(2010)** Denise Goya, Cleber Okida e Routo Terada. A two-party certificateless authenticated key agreement protocol. Em *SBSeg 2010 X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*. Sociedade Brasileira de Computação. Citado na pág. 53, 54

Granger *et al.***(2007)** R. Granger, F. Hess, R. Oyono, N. Thériault e F. Vercauteren. Ate pairing on hyperelliptic curves. Em *Proceedings of the 26th annual international conference on Advances in Cryptology*, EUROCRYPT '07, páginas 430–447, Berlin, Heidelberg. Springer-Verlag. ISBN 978-3-540-72539-8. doi: http://dx.doi.org/10.1007/978-3-540-72540-4_25. URL http://dx.doi.org/10.1007/978-3-540-72540-4_25. Citado na pág. 34

Gregorim *et al.***(2002)** Clóvis Osvaldo Gregorim, Creud Pereira Santos Martinelli e Sandra Helena Terciotti. *Michaelis : Dicionário escolar língua portuguesa*. Editora Melhoramentos Ltda., 6 edição. ISBN 8506034361. Citado na pág. 8

Hancock(2000) Harris Hancock. *Lectures on the theory of elliptic functions*. Dover Publications Inc., New York, NY, USA. ISBN 0486604837. URL http://www.archive.org/details/lecturestheorell00hancrich. Citado na pág. 15

Hess *et al.*(2006) F. Hess, N.P. Smart e F. Vercauteren. The eta pairing revisited. Cryptology ePrint Archive, Report 2006/110, 2006. http://eprint.iacr.org/2006/110. Citado na pág. 41, 65, 66, 67

Hess(2009) Florian Hess. Pairing lattices. Em *In Pairing 2009, volume 5209 of Lecture*. http://eprint.iacr.org/2008/125. Citado na pág. 42

Huang e Cao(2009a) Hai Huang e Zhenfu Cao. An id-based authenticated key exchange protocol based on bilinear diffie-hellman problem. Em *ASIACCS '09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, páginas 333–342, New York, NY, USA. ACM. ISBN 978-1-60558-394-5. http://eprint.iacr.org/2008/224. Citado na pág. 50

Huang e Cao(2009b) Hai Huang e Zhenfu Cao. An id-based authenticated key exchange protocol based on bilinear diffie-hellman problem. Em *ASIACCS '09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, páginas 333–342, New York, NY, USA. ACM. ISBN 978-1-60558-394-5. doi: http://doi.acm.org/10.1145/1533057.1533101. Citado na pág. 53

Ian F. Blake(1999) Nigel Paul Smart Ian F. Blake, Gadiel Seroussi. *Elliptic curves in cryptography*. London Mathematical Society, Lecture Note Series (No. 265). Cambridge University Press. ISBN 978-0521653749. doi: 10.2277/0521653746. URL http://www.cs.bris.ac.uk/~nigel/ECC/. Citado na pág. 15, 28

Icart(2009) Thomas Icart. How to hash into elliptic curves. Em *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, páginas 303–316, Berlin, Heidelberg. Springer-Verlag. ISBN 978-3-642-03355-1. doi: 10.1007/978-3-642-03356-8_18. URL http://portal.acm.org/citation.cfm?id=1615970.1615995. Citado na pág. 67, 76

institute of standards e technology(2002) National institute of standards e technology. Fips 180-2, secure hash standard, federal information processing standard (fips), publication 180-2. Relatório técnico, Department of Commerce. URL http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf. Citado na pág. 10

institute of standards e technology(2001) National institute of standards e technology. Descriptions of sha-256, sha-384, and sha-512. Relatório técnico, Department of Commerce. URL http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf. Citado na pág. 10

Jeffrey Hoffstein(2008) Joseph H. Silverman Jeffrey Hoffstein, Jill Pipher. *An Introduction to Mathematical Cryptography*. Undergraduate Texts in Mathematics. Springer-Verlag. ISBN 978-0-387-77993-5. Citado na pág. 15, 26

Joux(2009) Antoine Joux. *Algorithmic Cryptanalysis*. Chapman & Hall/CRC. ISBN 1420070029, 9781420070026. Citado na pág. 1

Katz e Lindell(2007) Jonathan Katz e Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC. ISBN 1584885513. Citado na pág. 1, 13

Koblitz(1987) Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177): 203–209. ISSN 0025-5718. doi: 10.1090/S0025-5718-1987-0866109-5. URL http://www.jstor.org/stable/2007884. Citado na pág. 27

Kohnfelder(1978) Loren M Kohnfelder. Towards a practical public-key cryptosystem. Citado na pág. 7

Krawczyk(2005) Hugo Krawczyk. Hmqv: A high-performance secure diffie-hellman protocol. Em *CRYPTO*, páginas 546–566. http://eprint.iacr.org/2005/176. Citado na pág. 13, 48, 49, 51

Kudla e Paterson(2005) Caroline Kudla e Kenneth G. Paterson. Modular security proofs for key agreement protocols. Em *ASIACRYPT*, páginas 549–565. URL http://www.iacr.org/cryptodb/archive/2005/ASIACRYPT/283/283.pdf. Citado na pág. 52

LaMacchia *et al.*(2007) Brian LaMacchia, Kristin Lauter e Anton Mityagin. Stronger security of authenticated key exchange. Em *ProvSec'07: Proceedings of the 1st international conference on Provable security*, páginas 1–16, Berlin, Heidelberg. Springer-Verlag. ISBN 3-540-75669-8, 978-3-540-75669-9. http://eprint.iacr.org/2006/073. Citado na pág. 47, 49, 50

Law *et al.*(2003) Laurie Law, Alfred Menezes, Minghua Qu, Jerry Solinas e Scott Vanstone. An efficient protocol for authenticated key agreement. Relatório técnico, Designs, Codes and Cryptography. http://www.springerlink.com/content/N84GX0461447N518. Citado na pág. 48

Lay e Zimmer(1994) Georg-Johann Lay e Horst Günter Zimmer. Constructing elliptic curves with given group order over large finite fields. Em *Proceedings of the First International Symposium on Algorithmic Number Theory*, páginas 250–263, London, UK. Springer-Verlag. ISBN 3-540-58691-1. URL http://portal.acm.org/citation.cfm?id=648182.749421. Citado na pág. 63

Libert e jacques Quisquater(2006) Benoît Libert e Jean jacques Quisquater. On constructing certificateless cryptosystems from identity based encryption. Em *In PKC 2006*, páginas 474–490. Springer-Verlag. Citado na pág. 47

Lippold *et al.*(2009) G Lippold, C Boyd e J.G. Nieto. Strongly secure certificateless key agreement. Em *Pairing 2009*, volume 5671 of *Lecture Notes in Computer Science*. Springer. Cryptology ePrint Archive, Report 2009/219, http://eprint.iacr.org/2009/219. Citado na pág. 4, 14, 47, 48, 49, 50, 51, 53, 54, 75

M. Joye(2008) G. Neven M. Joye. *Identity-Based Cryptography*. Volume 2 Cryptology and Information Security Series. IOS Press. ISBN 978-1586039479. URL http://www.iospress.nl/loadtop/load.php?isbn=9781586039479. Citado na pág. 33, 36, 40, 41, 42, 43, 44

Mandt e Tan(2006) T. K. Mandt e C. H. Tan. Certificateless Authenticated Two-Party Key Agreement Protocols. Em *11th Asian Computing Science Conference'06*, páginas 37–44. Springer Berlin. v.4435. Citado na pág. 14, 48, 49

Martin(2008) Luther Martin. *Introduction to Identity-Based Encryption*. Information Security and Privacy Series. Artech House Publishers. ISBN 978-1596932388. URL http://www.artechhouse.com/Detail.aspx?strIsbn=978-1-59693-238-8. Citado na pág. 5, 17, 19, 23, 24, 26, 32, 40

Menezes *et al.*(1996) Alfred J. Menezes, Scott A. Vanstone e Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA. URL http://www.cacr.math.uwaterloo.ca/hac/. Citado na pág. 12

Miller(1986) Victor S Miller. Use of elliptic curves in cryptography. Em *Lecture notes in computer sciences; 218 on Advances in cryptology—CRYPTO 85*, páginas 417–426, New York, NY, USA. Springer-Verlag New York, Inc. ISBN 0-387-16463-4. Citado na pág. 28

Miller(2004) Victor S. Miller. The weil pairing, and its efficient calculation. *Journal of Cryptology*, 17:235–261. ISSN 0933-2790. URL http://dx.doi.org/10.1007/s00145-004-0315-8. 10.1007/s00145-004-0315-8. Citado na pág. 32, 38

Miyaji *et al.*(**2001**) Miyaji, Nakabayashi e Takano. New Explicit Conditions of Elliptic Curve Traces for FR-Reduction. *TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems*. URL http://citeseer.ist.psu.edu/miyaji01new.html. Citado na pág. 64

M'Raïhi *et al.*(**1999**) David M'Raïhi, David Naccache, David Pointcheval e Serge Vaudenay. Computational alternatives to random number generators. Em *SAC '98: Proceedings of the Selected Areas in Cryptography*, páginas 72–80, London, UK. Springer-Verlag. ISBN 3-540-65894-7. Citado na pág. 11

Nielsen(2002) Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. Em *CRYPTO '02: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, páginas 111–126, London, UK. Springer-Verlag. ISBN 3-540-44050-X. Citado na pág. 11

Okamoto e Pointcheval(2001) Tatsuaki Okamoto e David Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. Em *PKC '01: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography*, páginas 104–118, London, UK. Springer-Verlag. ISBN 3-540-41658-7. Citado na pág. 12

Oppliger(2005) Rolf Oppliger. *Contemporary Cryptography (Artech House Computer Security503)*. Artech House Publishers. ISBN 1580536425. Citado na pág. 1

Oppliger(2009) Rolf Oppliger. *SSL and TLS: Theory and Practice*. Artech House, Inc., Norwood, MA, USA. ISBN 1596934476, 9781596934474. Citado na pág. 55

Pollard(1978) J. M. Pollard. Monte carlo methods for index computation (mod p). 32(143): 918–924. Citado na pág. 44

Sakai *et al.*(**2000**) R. Sakai, K. Ohgishi e M. Kasahara. Cryptosystems based on pairing. Em *Symposium on Cryptography and Information Security (SCIS2000)*, páginas 26–28, Okinawa, Japan. Inst. of Electronics, Information and Communication Engineers. Citado na pág. 50, 52

Scott(2007) Michael Scott. Implementing cryptographic pairings. Em Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto e Takeshi Okamoto, editors, *Pairing-Based Cryptography – Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, páginas 177–196. Springer Berlin / Heidelberg. Citado na pág. 62

Scott *et al.*(**2009**) Michael Scott, Naomi Benger, Manuel Charlemagne, Luis J. Dominguez Perez e Ezekiel J. Kachisa. Fast hashing to g2 on pairing-friendly curves. Em *Proceedings of the 3rd International Conference Palo Alto on Pairing-Based Cryptography*, Pairing '09, páginas 102–113, Berlin, Heidelberg. Springer-Verlag. ISBN 978-3-642-03297-4. doi: http://dx.doi.org/10.1007/978-3-642-03298-1_8. URL http://dx.doi.org/10.1007/978-3-642-03298-1_8. Citado na pág. 68

Shacham(2005) Hovav Shacham. *New Paradigms in Signature Schemes*. Tese de Doutorado, Stanford University. Citado na pág. 29

Shamir(1985) Adi Shamir. Identity-based cryptosystems and signature schemes. Em *Proceedings of CRYPTO 84 on Advances in cryptology*, páginas 47–53, New York, NY, USA. Springer-Verlag New York, Inc. ISBN 0-387-15658-5. doi: http://dx.doi.org/10.1007/3-540-39568-7_5. Citado na pág. 8

Shao(2005) Zuhua Shao. Efficient authenticated key agreement protocol using self-certified public keys from pairings. *Wuhan University Journal of Natural Sciences*, 10(1). Citado na pág. 14

Shim(2003) Kyungah Shim. Efficient id-based authenticated key agreement protocol based on weil pairing. *Electronics Letters*, 39(8):653 – 654. Citado na pág. 53

Shirey(2007) R. Shirey. Internet security glossary, version 2. RFC 4949 (Informational), Agosto 2007. URL http://www.ietf.org/rfc/rfc4949.txt. Citado na pág. 1, 2

Silverman(1986) Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, 1 edição. ISBN 0387962034. Citado na pág. 15, 17, 18, 24, 25

Stam e Lenstra(2003) Martijn Stam e Arjen K. Lenstra. Efficient subgroup exponentiation in quadratic and sixth degree extensions. Em *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '02, páginas 318–332, London, UK, UK. Springer-Verlag. ISBN 3-540-00409-2. URL http://portal.acm.org/citation.cfm?id=648255.752729. Citado na pág. 62

Stern *et al.*(2002) Jacques Stern, David Pointcheval, John Malone-lee e Nigel P. Smart. Flaws in applying proof methodologies to signature schemes. Em *In Advances in Cryptology crypto'02, Santa Barbara, Lectures Notes in Computer Science 2442*, páginas 93–110. Springer-Verlag. Citado na pág. 10

Swanson(2008) C. M. Swanson. Security in key agreement: Two-party certificateless schemes. Dissertação de Mestrado, University of Waterloo - Canadá. http://hdl.handle.net/10012/4156. Citado na pág. 13, 14, 47, 49

Talbot e Welsh(2006) John Talbot e D. J. A. Welsh. *Complexity and cryptography: an introduction*. Cambridge University Press. ISBN 978-0521852319. Citado na pág. 5

Taverne *et al.*(2011) Jonathan Taverne, Armando Faz-Hernández, Diego F. Aranha, Francisco Rodríguez-Henríquez, Darrel Hankerson e Julio López. Software implementation of binary elliptic curves: impact of the carry-less multiplier on scalar multiplication. Cryptology ePrint Archive, Report 2011/170, 2011. http://eprint.iacr.org/2011/170. Citado na pág. 75

Terada(2008) Routo Terada. *Segurança de Dados - Criptografia em Redes de Computador*. Editora Edgard Blücher, São Paulo, SP, 2 edição. Citado na pág. 72

Ustaoglu(2008) Berkant Ustaoglu. Obtaining a secure and efficient key agreement protocol from (h)mqv and naxos. *Des. Codes Cryptography*, 46(3):329–342. ISSN 0925-1022. doi: http://dx.doi.org/10.1007/s10623-007-9159-1. Citado na pág. 50

Vercauteren(2008) F. Vercauteren. Optimal pairings. Cryptology ePrint Archive, Report 2008/096, 2008. http://eprint.iacr.org/2008/096. Citado na pág. 42

Verheul(2004) Eric R. Verheul. Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. *Journal of Cryptology*, 17:277–296. ISSN 0933-2790. URL http://dx.doi.org/10.1007/s00145-004-0313-x. 10.1007/s00145-004-0313-x. Citado na pág. 19, 38

w. Bates *et al.*(2003) David w. Bates, Mark Ebell, Edward Gotlieb, John Zapp e H.C. Mullins. A proposal for electronic medical records in u.s. primary care. *Journal of the American Medical Informatics Association*, 10(1):1–10. Citado na pág. 3

Wang *et al.*(2003) Samuel J. Wang, Blackford Middleton, Lisa A. Prosser, Christina G. Bardon, Cynthia D. Spurr, Patricia J. Carchidi, Anne F. Kittler, Robert C. Goldszer andDavid G. Fairchild, Andrew J. Sussman, Gilad J. Kuperman e David W. Bates. A cost-benefit analysis of eletronic medical records in primary care. *The American Journal of Medicine*, 114(5):397–403. Citado na pág. 3

Wang e Yu(2005) Xiaoyun Wang e Hongbo Yu. How to break md5 and other hash functions. Em *In EUROCRYPT*. Springer-Verlag. Citado na pág. 10

Wang Shengbao(2006) Wang Licheng Wang Shengbao, Cao Zhenfu. Efficient certificateless authenticated key agreement protocol from pairings. *Wuhan University Journal of Natural Sciences*, 11. Citado na pág. 14

Washington(2008) Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Chapman & Hall/CRC, 2 edição. ISBN 9781420071467. Citado na pág. 21, 40, 64

Wayner(2008) Peter Wayner. *Disappearing Cryptography, Third Edition: Information Hiding Steganography & Watermarking*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3 edição. ISBN 0123744792, 9780123744791. Citado na pág. 2

Xia *et al.*(2008) Liang Xia, Shengbao Wang, Jiajun Shen e Guoming Xu. Breaking and repairing the certificateless key agreement protocol from asian 2006. *Wuhan University Journal of Natural Sciences*, 13(5):562–566. doi: 10.1007/s11859-008-0510-9. URL http://dx.doi.org/10.1007/s11859-008-0510-9. Citado na pág. 14, 48

Xiao e Chen(2008) Yang Xiao e Hui Chen. *Mobile Telemedicine: A Computing and Networking Perspective*. Auerbach Publications, Boston, MA, USA. ISBN 1420060465, 9781420060461. Citado na pág. 2