UMA PROPOSTA DE UTILIZAÇÃO DE CURVAS ELÍPTICAS NA CRIPTOGRAFIA BASEADA EM IDENTIDADES E SUA APLICAÇÃO NA TROCA SEGURA DE MENSAGENS

Julio Cesar Barbosa

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Luís Felipe Magalhães de Moraes, Ph. D.

Prof. Antonio Alberto Fernandes de Oliveira, D. Sc.

Prof. Reginaldo Palazzo Junior, Ph. D.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2005

BARBOSA, JULIO CESAR

Uma Proposta de Utilização de Curvas Elípticas na Criptografia Baseada em Identidades e sua Aplicação na Troca Segura de Mensagens [Rio de Janeiro] 2005

XIX, 145 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2005)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Segurança
2. Criptografia Assimétrica
3. Criptografia com Curvas Elípticas
4. Criptografia Baseada em Identidade
5. Simple Mail Transfer Protocol
6. Extensão SMTP

I. COPPE/UFRJ II. Título (série)

Agradecimentos

À minha esposa, por todo o suporte, carinho, dedicação e paciência.

Aos meus pais e irmãos, por todo o afeto, incentivo, força e tudo mais que possamos esperar de uma maravilhosa família.

Ao professor Luís Felipe M. de Moraes, pela orientação e apoio na elaboração deste trabalho.

Aos professores Antônio de Oliveira e Reginaldo Palazzo, por toda a contribuição.

A todos os amigos de trabalho, especialmente à Milu, Roberto Terra, Flávio Falcão, José Zava e Cristina Izzo, pelo incentivo, companheirismo e amizade.

Ao Projeto de Modernização das Fragatas, pela possibilidade de dedicação parcial que pôde viabilizar a realização deste trabalho.

Aos amigos Wanderley Lobianco e Marco Abcaran por todo o encorajamento e incentivo, fundamentais para o início dessa jornada.

Aos colegas do laboratório RAVEL Denilson Martins, Fernando Veríssimo, Carlos Alberto Vieira Campos, Daniel Otero, Luciano Renovato, Alexandre Pinaffi, Paulo Ditarso e aos demais companheiros desse e dos demais laboratórios da COPPE, pela troca de idéias e pela amizade que se formou ao longo destes anos.

Ao PESC e à COPPE, pelo suporte operacional.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UMA PROPOSTA DE UTILIZAÇÃO DE CURVAS ELÍPTICAS NA CRIPTOGRAFIA BASEADA EM IDENTIDADES E SUA APLICAÇÃO NA TROCA SEGURA DE MENSAGENS

Julio Cesar Barbosa

Abril/2005

Orientador: Luís Felipe Magalhães de Moraes

Programa: Engenharia de Sistemas e Computação

O objetivo deste trabalho é propor um modelo de *e-mail* seguro através da extensão do protocolo de transferência de mensagens SMTP - *Simple Mail Transfer Protocol*, implementada com base na criptografia de curvas elípticas e no uso de criptografia por identidades. Inicialmente, serão apresentados argumentos e motivações que justificam a utilização de curvas elípticas em ambientes com criptografia de chave assimétrica. Em seguida, será introduzida uma base teórica de criptografia por identidades, bem como suas vantagens e desvantagens em relação aos esquemas tradicionais de certificação digital. Como contribuição própria, o trabalho apresenta uma proposta de SMTP estendido com duas variações de implementação, adaptadas a partir de um servidor de *e-mails* real, que permitem a transferência de mensagens seguras. Por fim, comparações entre as duas implementações foram realizadas a fim de identificar qual o impacto no tempo médio de execução e também confrontar esses tempos com situações em que haja transferência insegura de mensagens.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A PROPOSAL OF USING ELLIPTIC CURVE IN IDENTITY BASED CRYPTOGRAPHY AND ITS APPLICATION FOR SECURE MESSAGE EXCHANGE

Julio Cesar Barbosa

April/2005

Advisor: Luís Felipe Magalhães de Moraes

Department: Systems Engineering and Computer Science

The main goal of this work is to propose a secure *e-mail* model as an SMTP - *Simple Mail Transfer Protocol* - extension, implemented based on both elliptic curve and identity cryptography. It begins with the explanation of the reasons for adopting elliptic curve cryptography in assymetric key environments. Afterwards, there will be introduced some theoretical basics for identity based cryptography as well as its advantages and disadvantages in comparison to traditional digital certificate schemes. As its own contribution, this text will introduce an extended SMTP proposal and two implemented versions of it, developed from a real and adapted *e-mails* server, that allows secure message transfer. At last, comparisons between these two implementations were made in other to identify the average time execution impact and also compare them to insecure message transfer scenarios.

Lista de Acrônimos

AC : *Autoridade Certificadora*;

ASCII : *American Standard Code for Information Interchange*;

AES : *Advanced Encryption Standard*;

API : *Application Programming Interface*;

BDHP : *Bilinear Diffie-Hellman Problem*;

CCE : *Criptografia com Curvas Elípticas*;

CDHP : *Computational Diffie-Hellman Problem*;

DDHP : *Decision Diffie-Hellman Problem*;

DES : *Data Encryption Standard*;

DLS : *Discrete Logarithm System*;

DLP : *Discrete Logarithm Problem*;

DSA : *Digital Signature Algorithm*;

ECC : *Elliptic Curve Cryptography*;

ECDLP : *Elliptic Curve Discrete Logarithm Problem*;

ECDLS : *Elliptic Curve Discrete Logarithm System*;

GCP : *Gerador de Chaves Privadas*;

GMP : *GNU Multiple Precision*;

GDHP : *Gap Diffie-Hellman Problem*;

HP : *Hewlett-Packard*;

IBE : *Identity Based Encryption*;

ICP : *Infra-estrutura de Chave Pública*;

IFP : *Integer Factorization Problem*;

IFS : *Integer Factorization System*;

IMAP : *Internet Message Access Protocol*;

IP : *Internet Protocol*;

MIPS : *Millions of Instructions Per Second*;

MUA : *Mail User Agent*;

MTA : *Mail Transfer Agent*;

OSI : *Open Systems Interconnection*;

PKG : *Private Key Generator*;

PKI : *Public Key Infraestructure*;

PGP : *Pretty Good Privacy*;

POP3 : *Post Office Protocol - Version 3*;

PLDCE : *Problema do Logaritmo Discreto em Curvas Elípticas*;

RSA : *Rivest Shamir Adleman*;

SCE : *Sistema de Curva Elíptica*;

SFI : *Sistema de Fatoração de Inteiros*;

SLD : *Sistema de Logaritmo Discreto*;

S/MIME : *Secure/Multipurpose Internet Mail Extensions*;

SMTP : *Simple Mail Transfer Protocol*;

SSL : *Security Sockets Language*;

RAM : *Random Access Memory*;

RFC : *Request For Comments*;

TCP : *Transmission Control Protocol*;

TLS : *Transport Layer Security*;

UCP : *Unidade Central de Processamento*;

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

DESDE os primeiros esforços no desenvolvimento de sistemas de segurança da informação, muitos estudos, nas mais diversas linhas de trabalho, têm sido feitos com o objetivo de alcançar modelos e projetos realmente eficientes nessa área. Motivos para tais investimentos não faltam, pois a rotina observada no dia-a-dia de uma sociedade moderna está cada vez mais ligada e dependente de estruturas que tornem tarefas, atualmente bastante comuns, como o acesso a extratos bancários e compras via *Internet*, minimamente seguras. Dessa forma, tem sido observado que a diversificação das opções de serviços baseados nessa grande rede necessita, com freqüência, de garantias de segurança para o seu funcionamento.

Tanto quanto o acesso a serviços hospedados em páginas da *Internet*, a viabilização do envio e recebimento de mensagens eletrônicas fez com que a troca de *e-mails* também fizesse parte da atual realidade tecnológica. Hoje em dia, o endereço eletrônico de uma conta de *e-mail* é quase tão necessário quanto um endereço de entrega de correspondências ou uma caixa postal, dado o seu elevado potencial como meio de comunicação, troca e compartilhamento de informações.

Contudo, assim como a maioria dos sistemas que se beneficiam da existência da *Internet* e da arquitetura de rede TCP/IP - *Transmission Control Protocol/Internet Protocol*, o correio eletrônico também é dependente de uma forte infra-estrutura de segurança. Conseqüentemente, da mesma forma que o comércio eletrônico necessita

de garantias fundamentais de segurança para o seu funcionamento, o serviço de *e-mail*, mesmo atendendo à sua funcionalidade básica de comunicação, também deve, idealmente, oferecer possibilidades de emprego que apresentem fatores como confidencialidade [1], autenticidade [2] e integridade [3].

O contínuo desenvolvimento de novas técnicas e estudos na área de criptografia [4] tem motivado a reflexão e reavaliação dos meios utilizados atualmente para oferecer segurança na troca de mensagens eletrônicas. Assim, é importante considerar essas novas possibilidades e, idealmente, criar meios de melhorar ainda mais esse serviço.

O restante deste capítulo se destina a apresentar a motivação que levou ao estudo de uma proposta de sistema de *e-mail* seguro, assim como o cenário em que este trabalho se aplica. O capítulo será, posteriormente, concluído com a organização estrutural contida nesta dissertação.

1.1 Motivação

É interessante identificar a crescente importância assumida pelo serviço de mensagens eletrônicas. Num mundo cada vez mais caracterizado pela velocidade com que informações são trocadas, a utilização de *e-mails* para comunicação rápida e eficiente entre pessoas, suporte a negócios, conectividade em ambientes profissionais e acadêmicos, entre muitas outras possibilidades de uso, passou a ser fundamental. Conforme foi mencionado no início deste capítulo, da mesma forma que diversos outros serviços disponibilizados através da *Internet*, a troca de mensagens eletrônicas também está exposta a problemas associados à segurança. A relativa facilidade com que ferramentas podem ser utilizadas na espionagem ou escuta eletrônica [5], muitas vezes atuando de forma "furtiva" e silenciosa, representam ameaças reais nesse contexto. Questões relacionadas a, por exemplo, falta de confidencialidade, podem

[1] Garantia de que somente o legítimo destinatário será capaz de ler uma mensagem.
[2] Garantia de que somente o verdadeiro remetente poderia ter gerado uma mensagem.
[3] Garantia de que o conteúdo de uma mensagem não foi alterado durante o seu transporte.
[4] Conjunto de técnicas que permitem converter um texto em cifras.
[5] Denominados "farejadores" ou *sniffers* de rede.

causar desde uma simples quebra de sigilo em uma mensagem de conteúdo pessoal, até graves prejuízos decorrentes de violações e roubo de informações envolvendo segredos de empresas ou entidades governamentais.

Ao se observar os recursos que "tradicionalmente" implementam *e-mail* seguro, constata-se a necessidade de uma maior convergência, em termos de aceitação, por parte da comunidade que se propõe a utilizar esse serviço. Nesse sentido, padronizações "de fato" dessas funcionalidades são, na prática, determinadas pela própria comunidade de usuários ou, quando a esses submetidas, dependem muito de sua adoção para que possam tornar-se realmente aceitas. Em outras palavras, de nada adianta a melhor e mais bem projetada infra-estrutura, se a mesma esbarra em não-aceitação por parte da comunidade para a qual a mesma foi concebida. Dessa forma, nada como a aceitação prática para promover ou condenar ao desuso, mesmo aquela que, teoricamente, seria a mais completa e promissora das propostas de *e-mail* seguro disponíveis.

Um dos principais motivos para as dificuldades de adoção enfrentadas pelas implementações "tradicionais" para troca de *e-mail* seguro está justamente na dificuldade de configuração, gerenciamento e conseqüente complexidade envolvida no seu funcionamento. Não obstante ao inquestionável êxito alcançado por esses mecanismos ao atingir os objetivos associados a garantia de segurança, seu desafio maior continua sendo superar essas dificuldades, que ainda inibem uma maior popularização da troca segura de mensagens eletrônicas.

Esforços no sentido de melhoria e simplificação do cenário de troca de mensagens seguras são, conseqüentemente, muito bem vindos e é justamente com essa motivação que este trabalho foi desenvolvido.

1.2 Cenário de Aplicação e Objetivo do Trabalho

Até um passado recente, as únicas técnicas para garantia de confidencialidade e autenticidade de mensagens eletrônicas baseavam-se na geração de chaves [6] aleatórias. Nesse contexto "tradicional", essas chaves são associadas às identidades dos seus portadores utilizando-se documentos eletrônicos digitalmente assinados, os chamados "certificados digitais". A gerência desses certificados, assim como a necessidade de geração dos mesmos, são obrigatórios de forma a, por exemplo, se poder codificar uma mensagem secreta para um destinatário. Esses são fatores que, inevitavelmente, aumentam a complexidade de funcionamento de todo o processo e, segundo algumas linhas de estudo de mercado [1], muito provavelmente, ajudam a justificar a carência de um padrão único.

Pesquisas [2] que viabilizassem a escolha de chaves (extraídas, por exemplo, a partir da própria identidade do portador) ao invés de serem geradas aleatoriamente, passaram, através de estudos mais recentes [3], a ser plenamente viáveis em termos práticos. Esses esforços têm se refletido em interessantes alternativas para aplicação dessas técnicas, apresentando tanto vantagens quanto desvantagens quando comparados às implementações "tradicionais", não somente de serviços de *e-mail* seguro, mas também nas mais diversas áreas relacionadas à segurança da informação (conforme será visto no decorrer deste texto). Conseqüentemente, pesquisas no sentido de oferecer uma simplificação no processo de troca de mensagens seguras têm surgido e já contam, inclusive, com implementações de mercado [4].

Esta dissertação visa propor uma alternativa que otimize ao máximo os procedimentos envolvidos na troca sigilosa de mensagens eletrônicas, utilizando, para esse fim, alguns dos mais recentes recursos e mais eficientes técnicas envolvendo criptografia, como criptografia com curvas elípticas [5] [6] e criptografia por identidades (ou IBE - *Identity Based Encryption*) [2] [3]. Assim, essas técnicas são utilizadas dentro de uma proposta de alteração do mais largamente adotado protocolo de troca de mensagens, o SMTP - *Simple Mail Transfer Protocol* [7] [8], dando origem, dessa forma, a uma alternativa que objetiva aliar segurança, simplicidade, viabilidade e

[6] Chaves são estruturas de dados utilizadas nas operações envolvendo técnicas criptográficas.

uma significativa facilidade de utilização e conseqüente potencial de adoção. Isto se deve ao fato de que estarão embutidos no protocolo SMTP, todos os passos de troca de parâmetros e chave de seção, necessários ao transporte de dados em formato criptografado no caminho entre o remetente e o destinatário final da mensagem. Assim, a todos os usuários que tivessem conta (e endereço) vinculada àquele domínio, estaria garantido o direito ao recebimento de mensagens sigilosas, sem a necessidade de cadastro de chaves públicas ou qualquer prévia negociação com um terceiro servidor ou entidade.

Além da proposta de mudança desse protocolo, realizada na forma de uma extensão SMTP (*SMTP Service Extension* [7]), uma implementação real dessa idéia também foi desenvolvida neste trabalho. Como complemento a esse desenvolvimento, medições foram realizadas de forma a avaliar o impacto e a viabilidade dessa idéia em termos práticos.

1.3 Descrição Estrutural do Trabalho

O Capítulo 2 descreve de forma mais abrangente o tema relacionado à criptografia, em especial criptografia de chave pública. Nesse momento, serão levantadas aplicações, métodos e algumas técnicas relacionadas, incluindo um comparativo entre as mesmas, onde estarão presentes os argumentos e vantagens advindas da criptografia de curvas elípticas.

No Capítulo 3 será melhor abordado como o emprego de curvas elípticas pode ser usado na geração de grupos e como técnica de criptografia. Será apresentada a aritmética de pontos em uma curva e como esses pontos são úteis em operações que sugerem o problema do logaritmo discreto em curvas elípticas (*Elliptic Curve Discrete Logarithm Problem - ECDLP*), base da segurança desse modelo de criptografia.

O Capítulo 4 tratará o tema relacionado a criptografia por identidades - *Identity*

[7] Propostas de alterações no padrão básico de funcionamento do protocolo SMTP, de acordo com o estabelecido em [9].

Based Encryption - IBE, introduzindo a teoria sobre os trabalhos de [2] e [3], somado a exemplos de aplicações de IBE em criptografia e IBE implementado em *e-mail* seguro. Assim como no Capítulo 2, esse capítulo também estará longe de apresentar um texto completo sobre esse assunto, logo, referências serão citadas com o intuito de complementar o entendimento sobre toda essa teoria.

O Capítulo 5 irá conter toda a proposta e contribuição deste trabalho. Ele está dividido em três níveis: apresentação do modelo de solução contido nesta dissertação; detalhamento da proposta de extensão (do protocolo de entrega de mensagens) sugerida e descrição da implementação da mesma. Na primeira parte, será apresentado não somente o modelo de solução idealizado, mas também decisões de projeto aplicadas no desenho dessa proposta. Num segundo momento, será descrita a extensão SMTP sugerida, com inclusão de novas sintaxes, comandos, etc, contidos nessa extensão. Além disso, serão apresentadas as ferramentas e recursos utilizados, bem como decisões tomadas e o relatório final da fase de implementação. Finalmente, também será incluída nesse capítulo uma avaliação do código implementado, contendo medições de desempenho e uma análise comparativa com cenários inseguros.

A conclusão do trabalho estará presente no Capítulo 6, onde serão apresentados comentários finais e trabalhos futuros, assim como melhorias e aprimoramentos que podem ser aplicados a essa proposta.

O Apêndice A contém algumas definições relativas a álgebra abstrata que serão utilizadas dentro desta dissertação.

O Apêndice B contém algumas definições relativas a corpos finitos. Essas definições são fundamentais para o funcionamento dos sistemas de criptografia que serão apresentados neste texto.

Capítulo 2

Criptografia

DENTRO do contexto de criptografia, a grande maioria dos sistemas existentes apoiam-se no fato de existirem problemas matemáticos que, dado o elevado nível de esforço envolvido na sua resolução, tornam-se de difícil solução. Conseqüentemente, busca-se proteção no fato de que o "adversário" não conseguirá, mesmo contando com as mais modernas ferramentas computacionais, reverter a função de criptografia (na qual o sistema se baseia) e acessar os parâmetros do sistema em um tempo aceitável.

Em termos de criptografia computacional, um problema matemático é dito "de difícil solução" quando, mesmo aplicando-se o algoritmo mais eficiente para resolvê-lo, esse leva um longo período de tempo para que sua execução se conclua. Esse tempo de execução possui uma relação direta com o tamanho dos dados de entrada do algoritmo utilizado. Cientistas da área [10] defendem o fato de que, em geral, um problema de fácil solução tem o tempo de execução polinomial, enquanto problemas de difícil solução tem esse tempo em formato exponencial. Dessa forma, há um grande interesse em saber o quanto um problema se torna difícil (tempo de execução) com o aumento do tamanho de sua entrada e, adicionalmente, em selecionar problemas que maximizem esse tempo, sempre que for necessária a obtenção de um sistema de criptografia mais seguro.

Nas próximas seções, será visto como o universo da criptografia pode ser di-

vidido em dois grandes grupos de técnicas: Criptografia Simétrica e Criptografia Assimétrica (ou Criptografia de Chave Pública). O restante do capítulo se dedicará a apresentar as mais diversas variações de métodos envolvendo criptografia de chave pública, que consiste no foco principal deste trabalho. Dessa forma, serão observados alguns comparativos e fatores relevantes a cada uma dessas técnicas.

2.1 Criptografia Simétrica

O advento de técnicas de criptografia em ambientes militares e acadêmicos proporcionou as bases para o desenvolvimento dos sistemas de segurança atualmente utilizados. Redes precursoras à hoje tão popular *Internet* já adotavam meios de criptografia modernos, ainda nos anos 70. Nesse sentido, um dos primeiros sistemas de criptografia que se popularizou é denominado de Sistema de Criptografia Simétrica.

Nos sistemas de criptografia simétrica, a mesma chave, ou, em alguns casos, duas chaves facilmente deriváveis uma da outra, são utilizadas tanto no processo de criptografia quanto na decriptografia. Assim, se uma mensagem cifrada irá trafegar entre as partes A e B, tanto A quanto B têm que concordar quanto a uma chave em comum a ser utilizada durante esse processo.

Como exemplo de sistema de criptografia simétrico, o DES - *Data Encryption Standard* - foi desenvolvido no início da década de 70 e adotado como padrão pelo governo americano em 1977 [11]. Mais recentemente, um novo padrão de criptografia simétrica, ainda mais robusto que o DES, foi proposto por Vincent Rijmen e Joan Daemen e igualmente adotado pelo governo americano: o AES - *Advanced Encryption Standard*. Esse novo sistema baseia-se no algoritmo Rijndael [12].

Exemplos de esquemas envolvendo criptografia simétrica são extremamente úteis e ainda muito necessários em cenários nos quais sigilo ou acesso a sistemas devam estar protegidos contra quebras de segurança. Esquemas como o Kerberos [13], onde um único servidor de chaves secretas compartilhadas é utilizado, permitem que os elementos envolvidos utilizem a mesma chave na troca de mensagens cifradas. A

Figura 2.1 exemplifica esse cenário.

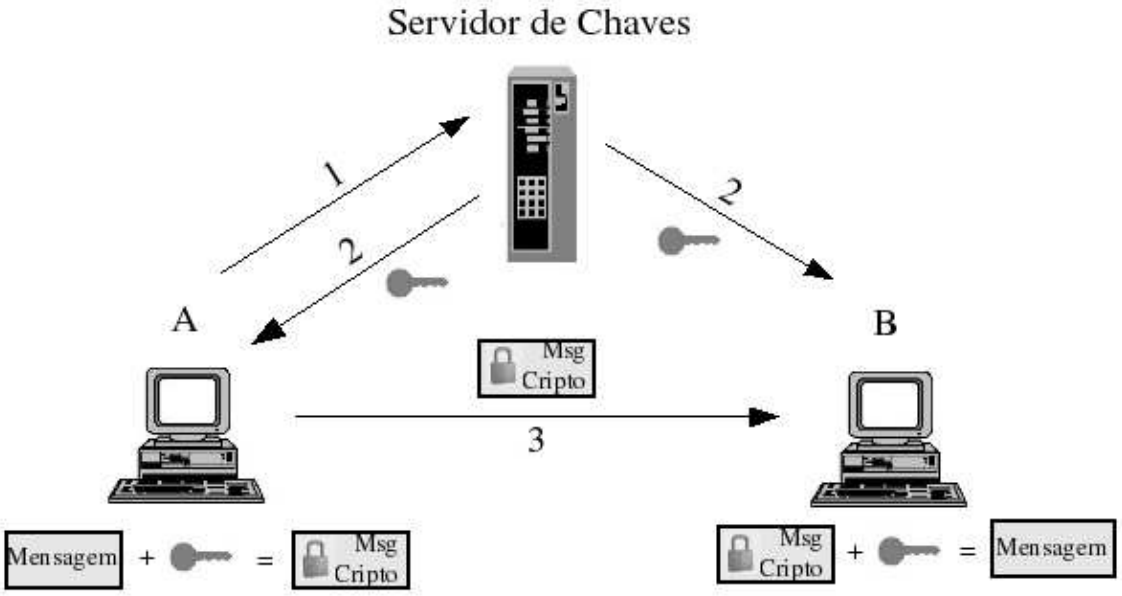

Figura 2.1: Criptografia Simétrica

Passo 1 A envia ao servidor de chaves uma requisição de senha (contendo o nome de B) criptografada;

Passo 2 O servidor de chaves gera, aleatoriamente, uma chave que é criptografada, respectivamente, com as chaves de A e B, e enviada a ambos;

Passo 3 A criptografa a mensagem com a chave gerada pelo servidor e envia o conteúdo cifrado a B que, por sua vez, tem a chave (simétrica) necessária à recuperação do texto original.

Alternativas de comunicação segura, como a demonstrada na Figura 2.1, dependem de um único sistema gerador de chaves que, infelizmente, fazem o uso desse tipo de abordagem esbarrar em aspectos que podem torná-la uma opção nem sempre atrativa. Dificuldades de escalabilidade e conseqüente sobrecarga advinda de múltiplas solicitações (múltiplos remetentes), assim como a dependência da presença de servidor e remetente conectados ao mesmo tempo devem, somente para citar alguns fatores, ser levados em consideração ao se projetar esse tipo de estrutura.

Esquemas alternativos para troca de chave secreta compartilhada sem a dependência de um servidor de chaves também podem ser aplicados dentro da criptografia simétrica. Essas alternativas, embora reduzam o número de participantes e simplifiquem bastante o cenário de compartilhamento de chaves, esbarram em outros tipos de problemas, como vulnerabilidades a ataques do tipo *man-in-the-middle* [14] e ausência de autenticação. Uma alternativa clássica pode ser vista no trabalho de Diffie e Hellman [15], que ainda serve de base para algumas implementações mais recentes, como pode ser visto no trabalho de [16].

2.2 Criptografia Assimétrica ou de Chave Pública

Uma grande variação aos sistemas de criptografia simétrica são os denominados sistemas assimétricos, ou de chaves assimétricas. Também chamados de sistemas de criptografia com chave pública, estes foram inicialmente propostos por Whitfield Diffie e Martin Hellman em 1976 [15]. Esses sistemas trabalham com duas chaves diferentes, independentes e não facilmente deriváveis [17]: A chave pública, que é utilizada na codificação de uma mensagem cifrada, e a chave privada, que é utilizada na sua decodificação [17][18].

A segurança desses sistemas está em poder armazenar a chave privada em segurança e ser computacionalmente impossível obter essa chave a partir da mensagem cifrada e/ou da chave pública correspondente [17]. A Figura 2.2 mostra como um cenário de troca segura de mensagem pode ser feito com o emprego de sistemas de chave pública.

Opcionalmente, quando se deseja garantir a autoria de uma mensagem, o emprego da chave pública e privada pode ser invertido. Assim, o remetente "assina" (codifica) digitalmente a mensagem através de sua chave privada, enquanto o destinatário somente conseguirá decodificar essa mensagem aplicando a chave pública do remetente. Contudo, esquemas de assinatura digital em sistemas de criptografia assimétrica, somente são viáveis nos casos em que os algoritmos empregados (alguns deles serão apresentados na Seção 2.4) possuam a propriedade de geração de um

texto cifrado, com a aplicação de uma chave privada, cuja versão original somente será recuperada aplicando-se a chave pública correspondente.

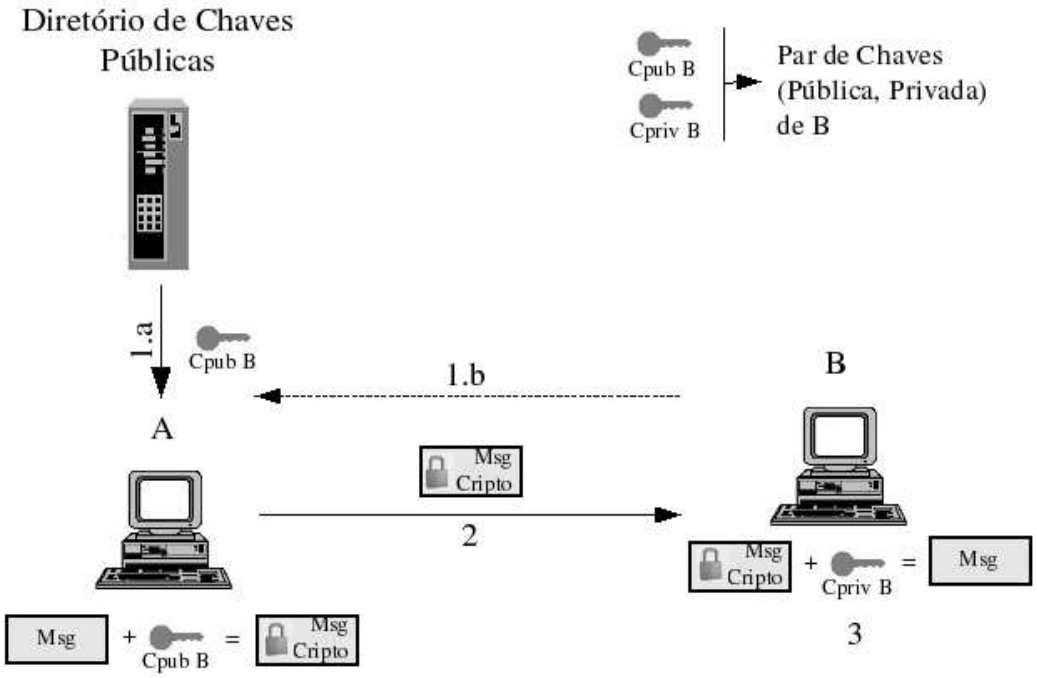

Figura 2.2: Criptografia Assimétrica ou de Chave Pública

Passo 1 A solicita a chave pública de B ($CpubB$) de um diretório específico de chaves (1.a) ou, opcionalmente, a partir do próprio B (1.b);

Passo 2 A criptografa o texto aplicando $CpubB$ e envia essa mensagem para B;

Passo 3 B abre o texto original aplicando sua chave privada $CprivB$ (que somente ele tem).

O funcionamento da assinatura digital é análogo ao caso em que se criptografa uma informação com uma chave pública, exceto pelo fato de que nem sempre o sigilo da informação é necessário. De fato, visando economia de banda e tempo de processamento, quando somente é desejada a comprovação de autoria (dispensando-se o sigilo), é possível executar funções de espalhamento (também conhecidas como funções de *hash*) no texto original e, enfim, aplicar a criptografia com chave privada no texto reduzido obtido dessa função. Assim, precisa ser garantido que essa função

de espalhamento somente produza um único resultado de tamanho fixo para um determinado tipo de informação, não importando o tamanho dessa informação inicial. Outra característica das funções de *hash* é o fato de ser impossível, a partir de um dado produzido por essa função, recuperar o dado que originalmente foi passado na entrada da mesma.

Pode-se observar, através da Figura 2.3, um esquema de assinaturas digitais que podem ser produzidas em sistemas de chave pública.

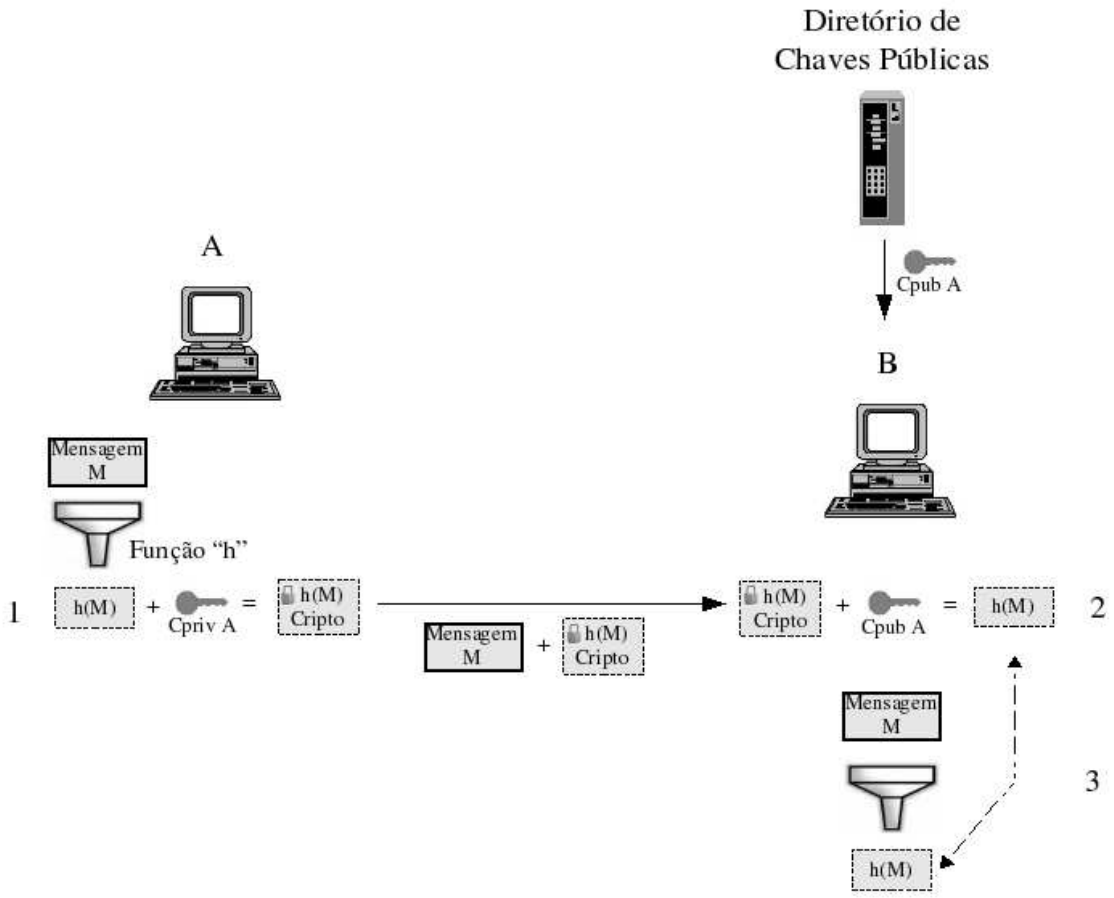

Figura 2.3: Assinatura Digital com Criptografia Assimétrica

Passo 1 A aplica sua chave privada $CprivA$ ao *hash* da mensagem e envia a mensagem e seu *hash* criptografado para B;

Passo 2 B solicita a chave pública de A ($CpubA$) de um diretório e, ao aplicá-la

ao *hash* criptografado recebido, recupera o *hash* original;

Passo 3 Ao obter o *hash* da mensagem de texto e compará-lo ao enviado com a mensagem, B certifica-se de que somente A poderia ter assinado tal texto.

Ao se comparar sistemas de chave privada com sistemas de chave pública, podem ser observadas vantagens e desvantagens, assim como diferenças relacionadas a potencialidade e aplicabilidade de cada uma dessas opções. Na prática, são utilizadas combinações do que há de melhor em cada uma dessas técnicas ao se projetar sistemas seguros. A Tabela 2.1 apresenta algumas das vantagens e desvantagens na adoção de cada um desses sistemas.

Sistema	**Vantagem/Desvantagem**
Simétrico	• Mais rápido e eficiente para grandes volumes de dados • Dificuldade para compartilhamento de chaves secretas • n pessoas precisam de $\binom{n}{2}$ chaves para se comunicar • Não possui comprovação ou negação da autoria de mensagens
Assimétrico	• Mais lento, restringindo-se a menores volumes de dados • Necessidade de infra-estrutura e gerenciamento de chaves • n pessoas precisam de somente $(2n)$ chaves para se comunicar • Possibilita a comprovação ou negação da autoria de mensagens • Necessidade de garantir que o possuidor da chave seja legítimo

Tabela 2.1: Vantagens e desvantagens entre criptografia simétrica e assimétrica

As próximas seções apresentam as aplicações de criptografia com chave pública, bem como as classes desses tipos de sistemas (incluindo criptografia com curvas elípticas) e comparações entre essas opções quanto à segurança e eficiência de processamento. Os argumentos e conclusões que serão apresentados a seguir, reforçarão o fato da criptografia de curvas elípticas ser uma excelente opção, não obstante um maior grau de complexidade para o seu entendimento.

2.3 Métodos e Técnicas para Sistemas de Chave Pública

Projetar algoritmos e funções que consigam criar um par de chaves totalmente distintas e de difícil derivação, mas que, em contrapartida, funcionem de forma eficiente, tanto do lado de quem envia o conteúdo (na Figura 2.2, A), quanto do lado de quem o recebe (B) não é, definitivamente, uma tarefa simples. Logo, tem se observado que, ao se projetar sistemas criptográficos de chave pública, é necessário haver um forte compromisso entre o nível de segurança [1] e o tempo de resposta [2] que se deseja obter [19]. Nesse aspecto, quanto mais desenvolvidos forem as ferramentas e algoritmos utilizados para violação dos sistemas de criptografia existentes, maiores têm que ser os parâmetros (chaves) e, conseqüentemente, maior o esforço no trabalho de codificação e decodificação dos textos cifrados. São nesses pontos que os métodos e técnicas de sistemas de chave pública devem ser avaliados e comparados entre si.

Muitas opções de sistemas de criptografia de chave pública já foram propostas, mas a maioria foi declarada comprovadamente insegura ou inviável em termos de uso prático, justamente por não atenderem aos requisitos descritos acima [10]. Atualmente, podem ser citados três principais tipos de sistemas de criptografia com chave pública considerados seguros e eficientes. Esses sistemas estão classificados de acordo com o problema matemático em que eles se baseiam:

- **Sistemas de Fatoração de Inteiros.** Os SFI (ou *Integer Factorization Systems - IFS*) são baseados no problema de fatoração de inteiros (*Integer Factorization Problem - IFP*) [10] [20];
- **Sistemas de Logaritmo Discreto.** Os SLD (ou *Discrete Logarithm System - DLS*) baseam-se no problema do logaritmo discreto (*Discrete Logarithm Problem - DLP*) [10] [20] [21];
- **Sistemas de Curva Elíptica.** Finalmente, os SCE (ou *Elliptic Curve Dis-*

[1] Esforço computacional necessário para violar um sistema.

[2] Tempo em que as funções que implementam o sistema são executadas.

crete Logarithm System - ECDLS) são baseados no problema do logaritmo discreto em curvas elípticas (*Elliptic Curve Discrete Logarithm Problem - EC-DLP*) [10] [20] [22] [23].

Os problemas em que esses sistemas de criptografia com chave pública se baseiam estão definidos de acordo com o apresentado pela Tabela 2.2.

Problema: Fatoração de inteiros	
Definição	Dado um número n que é o produto de dois valores primos grandes p e q (ou seja, $n = pq$), determinar p e q.
Base	Enquanto encontrar números primos grandes é uma tarefa relativamente fácil, o problema de fatorar o produto desses valores é considerado uma tarefa computacionalmente intratável [24].
Problema: Logaritmo discreto	
Definição	Dado um primo p, o conjunto $Z_p = \{0, 1, ..., p-1\}$, y, $g \in Z_p$, $g > 0$, determinar $x \in Z_p$, $0 \leqslant x \leqslant p-2$, onde $y \equiv g^x (\text{mod } p)$.
Base	O problema de calcular $x = logd_g(y)$ (onde $logd_g$ é o logaritmo discreto na base g), sendo g e y primos grandes, também é considerado uma tarefa computacionalmente intratável [21].
Problema: Logaritmo discreto em curvas elípticas	
Definição	Dada uma curva elíptica E, definida por um conjunto finito de pontos de natureza $\mathbb{F}_q$ (conforme definido no Apêndice B), onde q é o número de elementos desse conjunto (notação: $E(\mathbb{F}_q)$), e os pontos P, $Q \in E(\mathbb{F}_q)$, determinar o inteiro l, $0 \leqslant l \leqslant q-1$, tal que $Q = lP$.
Base	Ao passo que é relativamente fácil determinar o ponto $Q = lP$ (isso será visto ao definir as operações em curvas elípticas, no Capítulo 3), determinar l dados Q, $P \in E(\mathbb{F}_q)$ também é uma tarefa computacionalmente intratável [19].

Tabela 2.2: Características dos problemas na criptografia com chave pública

Exemplos de aplicações reais de cada uma dessas técnicas podem ser encontrados

em boa parte da literatura dedicada ao tema de criptografia assimétrica. Como ilustração de possíveis ferramentas relacionadas, a Tabela 2.3 lista variações em torno desse assunto, das quais algumas já são consagradas como padrão.

SFI	SLD	SCE
	ElGamal	
RSA	DSA	CCE
Rabin-Williams	Diffie-Hellman	IBE
	Schnorr	

Tabela 2.3: Exemplos de sistemas baseados em chave pública [10]

2.4 Revisão de Comparativos entre Sistemas Chave Pública

Esta seção pretende apresentar uma revisão de alguns resultados comparativos realizados entre sistemas de chave pública que podem ser encontrados na bibliografia relacionada a esse tema. Assim, além do estudo relativo às soluções para os problemas citados na Tabela 2.2, implementações dos mais populares [10] exemplos desses sistemas (Tabela 2.3), como RSA [25], DSA [26], ElGamal [21] e CCE [5] [6], foram confrontadas em trabalhos que serão referenciados no decorrer desta seção.

Ao se comparar os três sistemas de criptografia com chave pública, a maioria dos trabalhos abordam dois aspectos básicos nessa avaliação: segurança e eficiência. Outros aspectos não tão comumente discutidos são: aceitação pública, interoperabilidade e detalhamento técnico, que podem [10] [22] ser obtidos através da publicação de padrões desses sistemas.

2.4.1 Segurança

O escopo da análise de segurança em sistemas criptográficos com chave pública, limita-se na segurança "teórica", ou seja, em como violar esses sistemas. Discussões acerca de segurança em termos práticos, tais como confiança pessoal em funcionários ou responsáveis pelos sistemas, proteção e isolamento físico ou de acesso, políticas e medidas de segurança dentro das empresas, entre outros, embora sejam de grande relevância no contexto de segurança, não serão considerados nessa discussão.

A primeira pergunta ao se examinar a segurança de sistemas de criptografia com chave pública é: Violar o sistema requer, realmente, que seja resolvido o problema matemático em que ele se baseia? Segundo [10], uma prova matemática formal é fornecida com cada uma dessas propostas. Baseando-se nesse argumento, é assumido que a única forma de quebrar esses sistemas é por intermédio de algoritmos que tentem resolver, da forma mais eficiente possível, o problema proposto por cada um deles.

Os algoritmos propostos para resolver cada um dos problemas citados na Tabela 2.2, dividem-se em dois tipos [10]: algoritmos específicos (*special-purpose algoritms*) e algoritmos genéricos. Em termos gerais, os primeiros tratam somente alguns casos isolados, onde parâmetros específicos são utilizados nesses sistemas, enquanto os demais não fazem qualquer tipo de restrição quanto à parametrização utilizada.

Algoritmos específicos se baseiam em determinados aspectos de vulnerabilidade [3] que podem ser explorados em um conjunto restrito de situações. É razoável assumir que esses aspectos devem ser evitados ao se projetar os sistemas criptográficos. No caso da fatoração de inteiros, um algoritmo rápido pode ser projetado quando se utiliza fatores primos (p e q) "pequenos". De forma similar, o problema do logaritmo de números discretos pode ser resolvido quando são utilizados fatores primos "pequenos". Finalmente, no caso de curvas elípticas, duas pequenas classes de curvas denominadas supersingulares (*supersingular elliptic curves*) e anômalas (*anomalous elliptic curves*) também apresentam aspectos de vulnerabilidade e, conseqüente-

[3] Característica que torna o sistema vulnerável a um determinado algoritmo de quebra, aplicável somente naquele caso específico.

mente, também podem ser tratadas por algoritmos específicos [22].

Os algoritmos genéricos, ao contrário, se propõem a resolver qualquer configuração encontrada em cada um desses problemas. Independente dos parâmetros utilizados, esses algoritmos sempre chegam à resposta [10]. A questão agora é: Quanto tempo esses algoritmos levam para encontrar a solução? A ordem de grandeza desse tempo irá permitir avaliação de, dados os parâmetros de entrada, quanto um sistema de criptografia com chave pública é seguro.

Ao avaliar e comparar as opções de sistemas de criptografia com chave pública, os cientistas da área se baseiam nos algoritmos genéricos e qual a complexidade (número de passos X tamanho da entrada) que cada um deles oferece. Os problemas de fatoração de inteiros e de logaritmos discretos admitem, em geral, algoritmos que executam em tempo sub-exponencial [10]. Esses problemas também são considerados "difíceis", mas não tão difíceis quanto os que necessitam de algoritmos puramente exponenciais. Por outro lado, o melhor algoritmo genérico para o problema dos logaritmos discretos em curvas elípticas é puramente exponencial [22].

Baseando-se nos valores de complexidade obtidos em cada caso, pode-se observar que o problema de logaritmos discretos em curvas elípticas é considerado mais "difícil" de resolver que os demais. Como exemplo concreto dessa superioridade, a Figura 2.4 apresenta o tempo necessário para violar um sistema CCE em comparação com as aplicações RSA e DSA. Esse esforço para quebra dos sistemas está apresentado em unidades de MIPS (número de anos que uma máquina, capaz de executar um milhão de instruções por segundo, leva para resolver o problema). Vale ressaltar que, uma medida atualmente considerada como parâmetro "razoável" para o nível de segurança é 10^{12} MIPS (10^{12} anos) [20].

Pode ser observado através da Figura 2.4, o fato de que, para um nível de segurança razoável (10^{12} MIPS), enquanto o RSA e o DSA necessitam de 1024 bits, o CCE precisa de somente 160 bits para o tamanho de chave. Um outro fato interessante é que o aumento do nível de segurança (MIPS maior) necessita de um aumento bem mais expressivo do tamanho das chaves do RSA e DSA, em comparação ao CCE. Isso evidencia que o aumento dos atuais parâmetros de segurança, irá

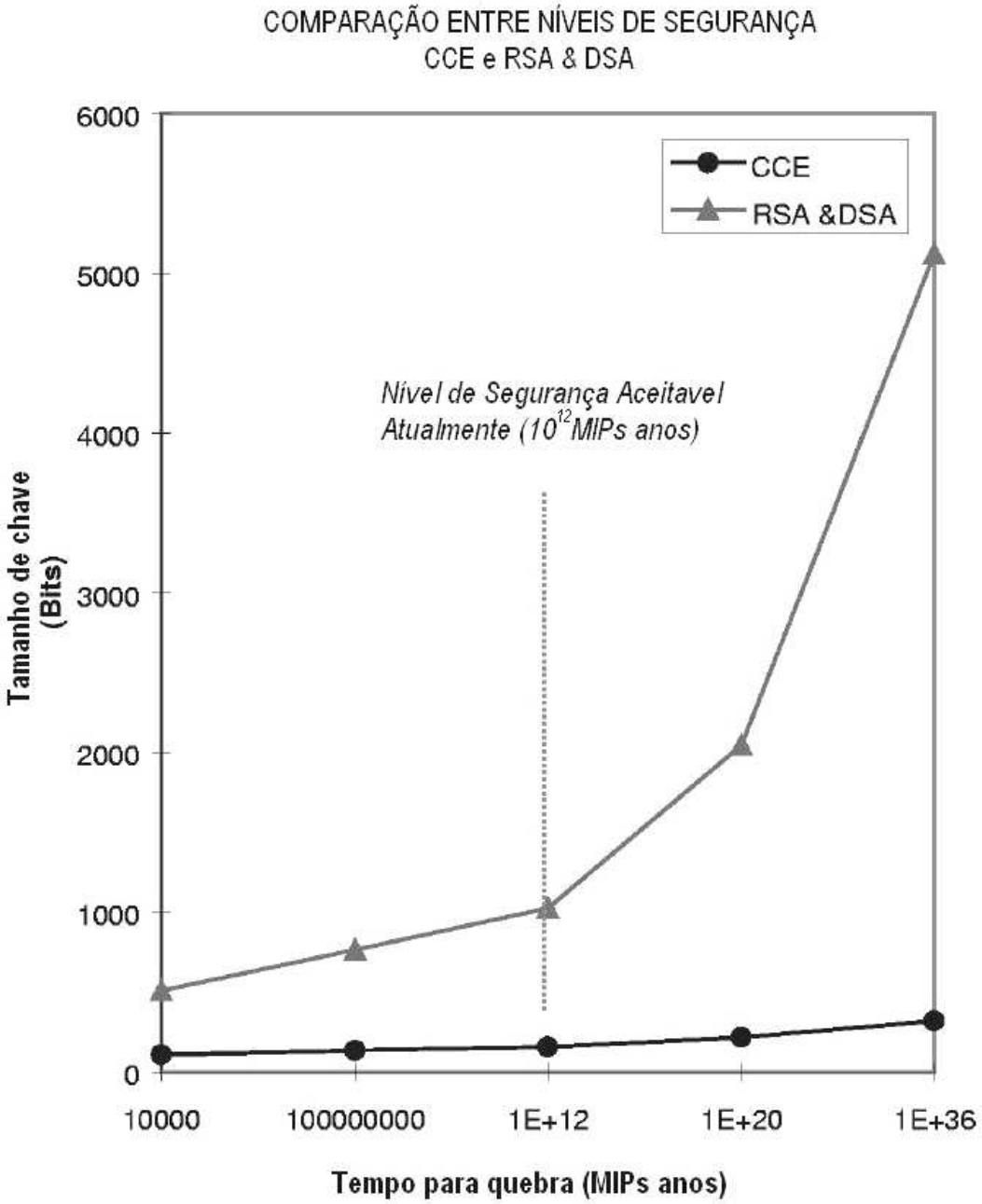

Figura 2.4: Segurança X Tamanho de chave (CCE, RSA e DSA) [10]

exigir um crescimento do tamanho da chave bem mais significativo no caso do RSA e DSA do que no CCE.

Retirada a partir do trabalho de [27], a Tabela 2.4 também mostra como algumas dessas implementações podem serem organizadas em tamanho de chave para um nível de segurança equivalente.

CCE	RSA/DSA	MIPS ano para quebra
160	1024	10^{12}
224	2048	10^{24}
256	3072	10^{28}
384	7680	10^{47}
512	15360	10^{66}

Tabela 2.4: Tamanho de chave para um mesmo nível de segurança [27]

A Seção 2.4.2, que prossegue o trabalho comparativo em termos de eficiência, voltará a abordar os aspectos referentes ao tamanho de chave entre essas opções de sistemas criptográficos.

2.4.2 Eficiência

A discussão acerca da eficiência de cada um dos sistemas de criptografia descritos aqui leva em consideração os seguintes fatores: Carga computacional, tamanho de chave e tamanho de banda. Para uma comparação mais justa, os dados apresentados nesta seção levam em consideração o mesmo nível de segurança para todas as propostas analisadas (CCE, RSA ou DSA).

Carga Computacional. Mede a eficiência com que os algoritmos podem implementar as transformações com as chaves públicas e privadas (sistema em operação). Segundo [10], as melhores implementações de cada um dos sistemas (*"state-of-the-art implementations"*) indicam que o CCE executa, aproximadamente, 10 vezes mais rápido que o RSA ou DSA.

Como exemplo prático dessa vantagem, o trabalho de [27] mediu a melhoria no tempo de conexão em transferências HTTPS [4], utilizando páginas *Web* de tamanhos diferentes e adotando implementações RSA e CCE com diferentes

[4] Conexões seguras para transferência de páginas da *Internet*.

tamanhos de chave. A Figura 2.5, cuja origem é [27], detalha os resultados das medições realizadas.

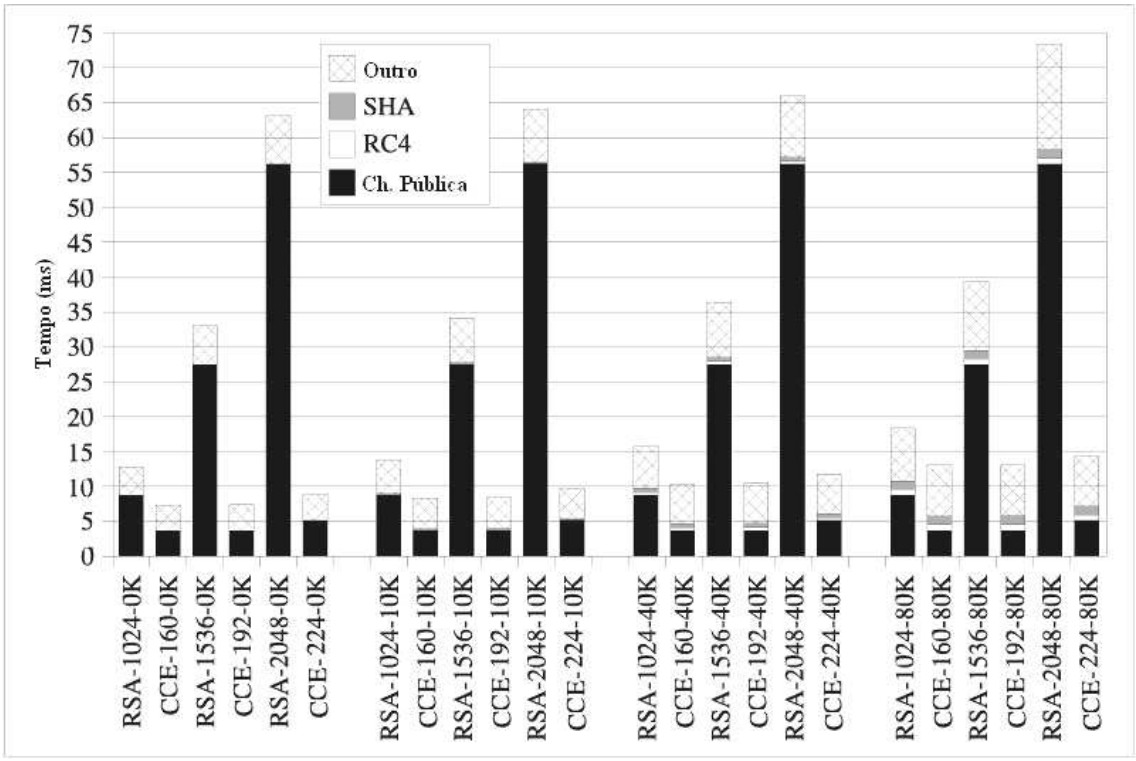

Figura 2.5: Tempos de Transmissão HTTPS (CCE e RSA) [27]

Tamanho de Chave. Conforme visto na Seção 2.4.1, o CCE também apresenta grande vantagem nesse aspecto. A Tabela 2.5 consolida essa vantagem.

	Parâmetros do Sistema	**Chave Pública (bits)**	**Chave Privada (bits)**
RSA	n/a	1088	2048
DSA	2208	1024	160
CCE	481	161	160

Tabela 2.5: Tamanho dos parâmetros e par de chaves [10]

Tamanho de Banda. Mede quantos bits a mais têm que ser transmitidos após criptografar ou assinar uma mensagem, em relação à mensagem original. Todas as três opções apresentam valores parecidos nesse quesito, com o CCE se destacando, exclusivamente, nos casos em que deseja-se processar mensagens pequenas, conforme será verificado nas tabelas a seguir. Ao se visualizar sistemas de criptografia com chave pública como uma eficiente ferramenta para troca de chave de seção (usa transformação de mensagens pequenas), essa vantagem do CCE fica ainda mais significativa. As Tabelas 2.6 e 2.7 demonstram como o CCE torna-se, isoladamente, a melhor opção ao se transformar textos pequenos (100 bits) em relação às mensagens mais longas (assinaturas de 2000 bits de comprimento).

	Tamanho da Assinatura (bits)
RSA	1024
DSA	320
CCE	320

Tabela 2.6: Assinatura de mensagens longas (2000 bits) [10]

	Mensagem Codificada (bits)
RSA	1024
ElGamal	2048
CCE	321

Tabela 2.7: Criptografia de mensagens curtas (100 bits) [10]

Definitivamente, uso da criptografia com chave pública baseada em curvas elípticas é uma excelente opção, não somente em termos de nível de segurança, como também em todos os principais pontos relativos à eficiência de operação. O fato desses sistemas utilizarem chaves menores, sem perda do nível de segurança, motiva

ainda mais o uso dessa técnica em sistemas com restrições de recursos (memória, CPU, etc).

Por se tratar de uma ferramenta fundamental na confecção deste trabalho, o próximo capítulo pretende apresentar essa técnica de forma mais detalhada.

Capítulo 3

Curvas Elípticas e Criptografia

O estudo de curvas elípticas como uma classe de curvas cúbicas e não-singulares [23] é bem anterior a sua aplicação em criptografia. Segundo [22], curvas elípticas têm sido extensivamente estudadas por mais de cem anos, havendo uma vasta literatura relacionada a esse tema. Mais recentemente, o estudo de curvas elípticas tem sido aplicado às mais diversas áreas, como geração de números pseudo-aleatórios [28] e ferramentas para teste de primalidade [29], entre outros exemplos de uso [23].

Em 1985, Victor Miller [5] e Neal Koblitz [6] sugeriram, de forma independente, uma técnica que utiliza as propriedades dessas curvas de forma a produzir um atraente sistema de criptografia baseado em chave pública. Essa técnica apresenta dois aspectos básicos que muito motivam sua aplicação [23]: operações entre elementos de grupo (nesse caso, pontos) com poucas operações aritméticas - oferecendo um ótimo desempenho a baixo custo, aliadas ao problema do logaritmo discreto em curvas elípticas que, conforme visto no Capítulo 2, dentre as funções matemáticas que atualmente implementam criptografia de chave pública, é a que envolve o maior custo para sua inversão.

Antes de falar mais especificamente sobre a aplicação de curvas elípticas em criptografia, as próximas seções irão apresentar, de forma resumida, o que são curvas elípticas e quais são suas características básicas. Posteriormente, será abordada a utilização dessas curvas em ambientes próprios à sua aplicação em sistemas cripto-

gráficos e, ao final deste capítulo, será mostrado um exemplo de uso dessa técnica.

3.1 Álgebra de Curvas Elípticas

Esta seção propõe-se a introduzir alguns conceitos básicos sobre curvas elípticas, suas propriedades e a álgebra envolvida, denominada como álgebra de curvas elípticas. Não haverá a pretensão, nesse texto, de detalhar aspectos matemáticos envolvendo esses tipos de curvas, visto que esse não é o escopo principal deste trabalho. Dessa forma, essa abordagem irá se concentrar em aspectos necessários a um nível de entendimento que proporcione a visualização desse recurso no campo de aplicações de criptografia. Referências serão deixadas como importantes complementos necessários a um entendimento mais completo desse tema.

É importante frisar que curvas elípticas não são elipses. Elas têm esse nome pois são definidas como um objeto matemático (uma curva) descrito por uma equação cúbica [17], as mesmas usadas para calcular o comprimento de arco de uma elipse [30]. Essas curvas, que podem assumir diversas formas (dependendo dos parâmetros utilizados), possuem propriedades interessantes [31] e o interesse nelas está justamente nessas propriedades. Em particular, podem ser definidas, a partir do conjunto de soluções (pontos) dessa curva, operações específicas e um elemento identidade, conforme será visto mais adiante.

Equações cúbicas para curvas elípticas têm a seguinte forma geral [32]:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_5 \tag{3.1}$$

Inicialmente, será assumido que os valores de x, y e a_i sejam números reais. De forma mais genérica, eles poderiam ser valores inteiros, complexos, base canônica ou qualquer outro tipo de elemento de um corpo [32]. Esta seção irá se ater à representação no plano dos reais, onde visualização dessas curvas é mais natural.

A Equação 3.2 [32] é uma forma mais simplificada da Equação 3.1:

$$y^2 = x^3 + a_4x + a_5 \tag{3.2}$$

Ao desenhar uma curva a partir da Equação 3.2 com os parâmetros $a_4 = -4$ e $a_5 = 0.67$, será obtido o gráfico da Figura 3.1.

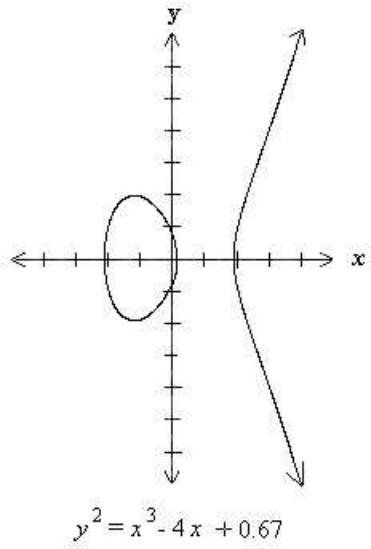

Figura 3.1: Exemplo de curva elíptica [31]

Pode se definir "soma" de dois pontos pertencentes a uma curva elíptica (álgebra para curvas elípticas, [32]) como sendo um terceiro ponto, também presente na mesma. Um elemento identidade (citado anteriormente) nessa álgebra seria um ponto O - chamado de "ponto no infinito", tal que a soma de qualquer outro ponto da curva a esse ponto resulta no próprio ponto. Isso seria equivalente à soma de um inteiro 0 com 0, na álgebra tradicional.

Dessa forma, dada uma curva elíptica E e os pontos P, $O \in E$:

$$O = -O \tag{3.3}$$

$$P + O = P \tag{3.4}$$

Geometricamente, a "soma elíptica" de dois pontos P e Q presentes em uma

curva elíptica é obtida através de uma reta que atravesse esses dois pontos, atravessando também um terceiro ponto dessa curva, que representa o resultado dessa soma "rebatido" no eixo horizontal. Também em termos gráficos, o ponto O está localizado em um lugar infinitamente distante, sobre o eixo vertical. A Figura 3.2 representa o ponto resultante R da "soma" de P e Q.

$$R = P + Q \tag{3.5}$$

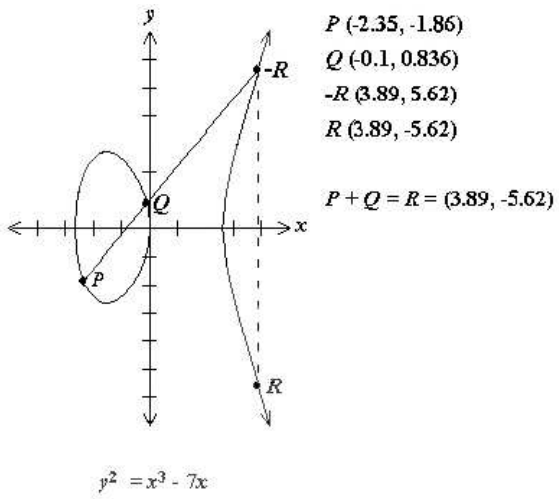

Figura 3.2: Soma elíptica ($R = P + Q$) [31]

Adicionalmente, pode-se representar as operações de soma de um ponto com ele mesmo ("dobrar" o ponto) como um ponto "rebatido" no eixo horizontal, a partir do ponto de interseção entre a reta tangente ao ponto que se deseja "dobrar" e a curva. A soma de um ponto com seu "negativo", assim como a "dobra" de um ponto $P = (x_p, y_p)$, onde $y_p = 0$, levam ao ponto O. As próximas figuras exemplificam esses três casos. Matematicamente, tem-se que:

$$R = 2P = P + P \tag{3.6}$$

$$P = (x_p, y_p) + [(-P) = (x_p, -y_p)] = O \tag{3.7}$$

$$P = (x_p, y_p) \forall y_p = 0 \Rightarrow 2P = O \tag{3.8}$$

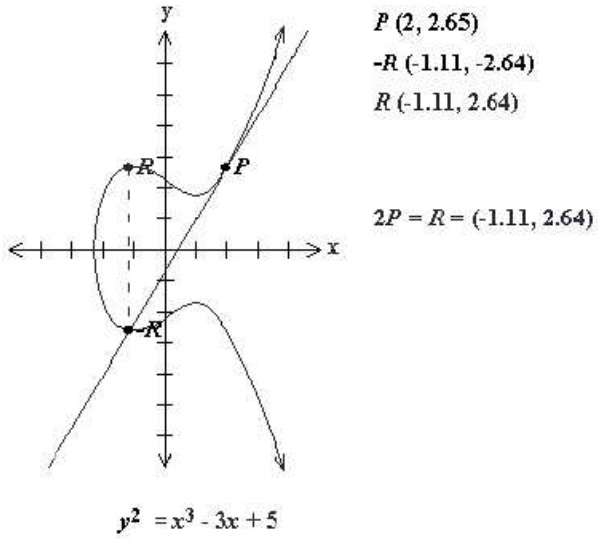

Figura 3.3: Soma de um mesmo ponto ($R = P + P = 2P$) [31]

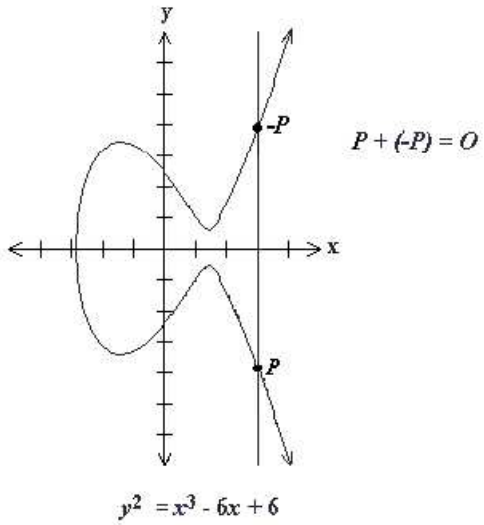

Figura 3.4: Soma de um ponto com seu oposto ($P + (-P) = O$) [31]

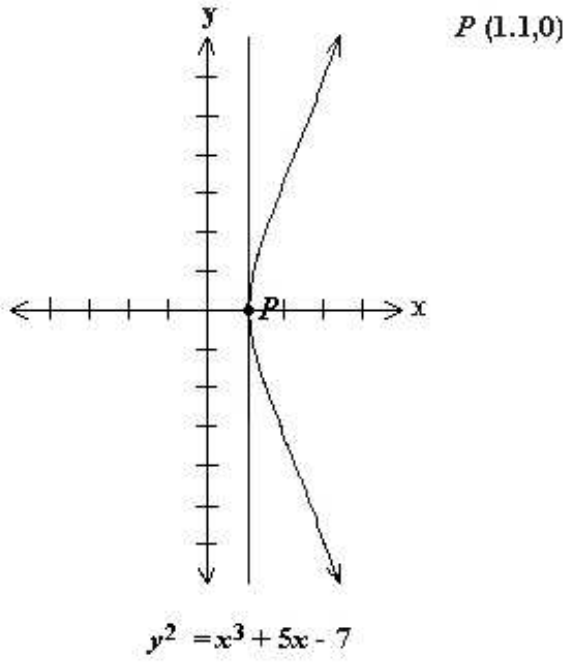

Figura 3.5: Dobro de um ponto (x_p, y_p), onde $y_p = 0$ [31]

A multiplicação de um ponto em uma curva elíptica por um valor inteiro qualquer (maior que 2), segue a mesma forma da Equação 3.6, ou seja, basta somar o ponto a ele mesmo tantas vezes quanto for o fator de multiplicação. Dessa forma, observa-se que:

$$R = lP = \overbrace{P + P + ... + P}^{l} \tag{3.9}$$

A título de exemplo [32], a partir da Figura 3.3, uma vez havendo interesse em calcular $Q = 15P$, bastaria reaplicar, continuamente, as operações de soma e soma de mesmo ponto, de forma a obter o valor desejado (também um ponto na curva):

$$Q = 15P = P + 2(P + 2(P + 2P)) \tag{3.10}$$

De forma resumida, operações de soma de pontos de uma curva elíptica seguem as seguintes regras para todos os pontos P, $Q \in E$ [23]:

1. $O + P = P$ e $P + O = P$

2. $-O = O$

3. Se $P = (x1, y1) \neq O$, então $-P = (x1, -y1)$

4. Se $Q = -P$, então $P + Q = O$

5. Se $P \neq O$, $Q \neq O$, $Q \neq -P$, então, seja R o ponto de interseção entre a curva e a linha $\overline{PQ}$ (caso $P \neq Q$) ou a linha tangente a P (caso $P = Q$). Então $P + Q = -R$

Graficamente, esse conjunto de operações está representado na Figura 3.6:

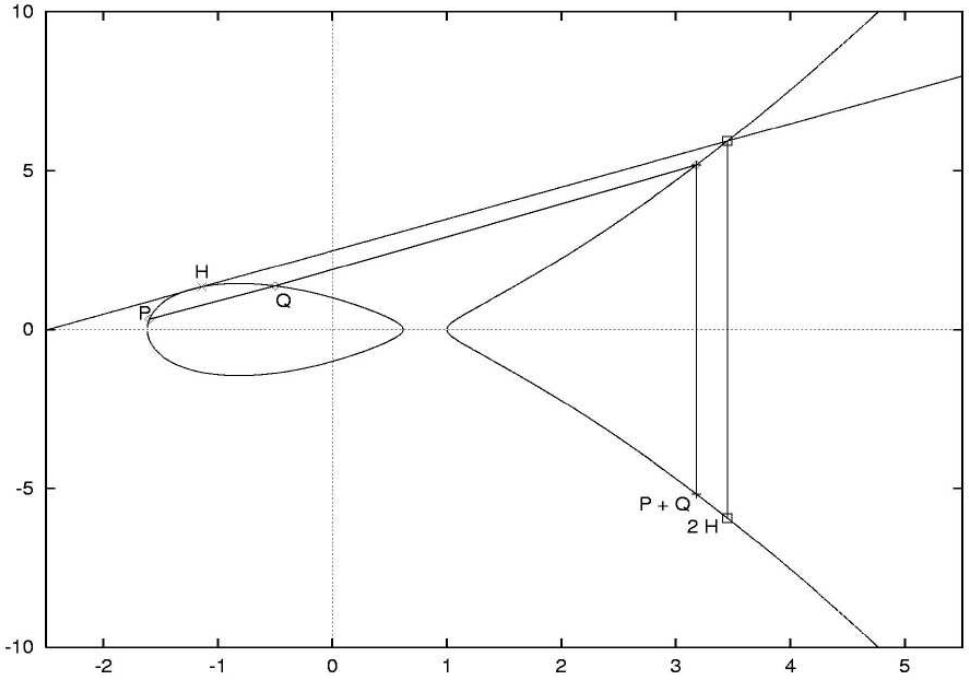

Figura 3.6: Exemplos de soma elíptica de pontos [19]

Adição e multiplicação de pontos de uma curva elíptica são as principais operações feitas nessas curvas. Essas operações podem ser implementadas tanto em *hardware* quanto em *software* e existem algoritmos matemáticos bastante eficientes para realizá-las [33]. Essa característica confere à criptografia com curvas elípticas uma boa opção também quanto ao desempenho de operação, conforme citado na Seção 2.4.2.

Ao aplicar essas operações de forma restrita a um grupo finito de inteiros, os proponentes da criptografia com curva elíptica observaram todo o potencial dessa técnica. Conforme citado na Tabela 2.2, eles se basearam no fato de que, embora seja relativamente fácil determinar o ponto $Q = lP$, determinar o inteiro l dados Q, $P \in E(\mathbb{F}_q)$ (E é uma curva elíptica no grupo $\mathbb{F}_q$) é bem mais difícil e não possui solução em tempo sub-exponencial.

3.2 Aplicação de Curvas Elípticas em Criptografia

Embora as figuras apresentadas até agora tenham nos familiarizado com o uso de curvas elípticas aplicadas ao plano dos reais, seu uso em sistemas criptográficos seria inviável. Isso se deve ao fato de que, ao utilizar corpos infinitos, como o conjunto de reais ou complexos, ter-se-ia problemas com arredondamento, "trucagem" de valores e limites [31]. Soma-se a isso o fato de sistemas de criptografia necessitarem de uma aritmética rápida e precisa, o que pode ser obtido através dos também denominados "corpos finitos", cuja notação é: $\mathbb{F}_q$.

Voltando à equação genérica de curvas elípticas 3.1, a aplicação de curvas elípticas em criptografia utiliza valores de x e y (que satisfazem à mesma) pertencentes a $\mathbb{F}_q$. Assim, o grande mérito dessas curvas aplicadas a corpos finitos (ex.: conjunto finito de inteiros [10]) é possibilitar a criação de um grupo fechado e numeroso, porém finito, de pontos sob os quais todas as operações vistas até agora (soma, produto, etc) se mantêm.

De acordo com [23], os pontos que satisfazem a equação de uma curva elíptica $E(\mathbb{F}_q)$ formam um grupo chamado "grupo abeliano" (definido também no Apêndice A). Nesse grupo, podem ser definidas operações de soma que levam, obrigatoriamente, a outros pontos dentro desse mesmo grupo. Somas sucessivas de um mesmo ponto, são tratadas como produto de um escalar k pelo ponto P que está sendo, sucessivamente, somado k vezes: $Q = P + P + ... + P$ (k vezes) $= kP$. Nesse contexto, não existe nenhum algoritmo eficiente que, dado Q e P, forneça o valor de k . De acordo com o que foi visto no Capítulo 2, esse problema é conhecido

como Problema do Logaritmo Discreto em Curvas Elípticas [10] [20] (*PLDCE*) e é justamente nesse argumento que se baseia a segurança da criptografia por curvas elípticas.

A partir desse momento, esse texto passará a considerar somente curvas elípticas definidas sobre um conjunto finito de valores, ou "corpos finitos", $\mathbb{F}_q$. A definição e aplicação de curva elíptica nesses corpos será melhor abordada nas próximas seções.

3.2.1 Curvas Elípticas sobre Corpos Finitos

Duas das mais eficientes e freqüentemente utilizadas opções de $\mathbb{F}_q$ são [10][31]: Curvas elípticas sobre corpos finitos primos ($\mathbb{F}_q = \mathbb{F}_p$) e curvas elípticas sobre corpos finitos de característica dois ($\mathbb{F}_q = \mathbb{F}_{2^m}$). Essas curvas também são denominadas, respectivamente, curvas em corpos de característica p e curvas em corpos de característica 2 [33]. As duas opções são tratadas, separadamente, nas próximas seções.

3.2.2 Curvas Elípticas sobre Corpos Finitos Primos

O conjunto $\mathbb{F}_p$ é composto por valores de 0 a $p-1$ ($\mathbb{F}_p = \{0, 1, ..., p-1\}$) e todas as operações devem ser finalizadas calculando-se o resto da divisão por p. Dessa forma, sempre são alcançados resultados dentro do conjunto formado por $\mathbb{F}_p$.

Uma curva elíptica sobre o corpo $\mathbb{F}_p$ é definida, a partir da Equação 3.2, pela equação a seguir [34]:

$$(y^2) \mod p = (x^3 + ax + b) \mod p \tag{3.11}$$

onde a, $b \in \mathbb{F}_p$. A curva elíptica inclui todos os pontos (x, y) que satisfazem a Equação 3.11, mais o ponto O, e x, $y \in \mathbb{F}_p$.

Adicionalmente, é importante citar que, para garantir que $(x^3 + ax + b)$ não possua fatores repetidos (a curva realmente forme um grupo), deve-se obedecer à seguinte restrição [17][31][34]:

$$(4a^3 + 27b^2 \mod p) \neq 0 \tag{3.12}$$

As equações a seguir definem as operações de soma de pontos numa curva elíptica sobre $\mathbb{F}_p$ [33][31]:

$$P_3(x_3, y_3) = P_1(x_1, x_1) + P_2(x_2, y_2), \text{ onde } P_3 \neq O$$

$$x_3 = (\lambda^2 - x_1 - x_2) \mod p \tag{3.13}$$

$$y_3 = ((x_1 - x_3)\lambda - y_1) \mod p \tag{3.14}$$

onde:

$$x_1 \neq x_2 \Rightarrow \lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod p$$

$$x_1 = x_2 \ \forall \ y_2 \neq 0 \Rightarrow \lambda = \frac{3x_1^2 + a}{2y_1} \pmod p$$

Exemplo de Curvas Elípticas sobre $\mathbb{F}_p$ ($\mathbb{F}_{23}$) [34]:

Considerando uma curva elíptica $E(\mathbb{F}_{23})$ com a equação $y^2 = x^3 + x$ ($a = 1$ e $b = 0$), pode ser observado que o ponto $(9, 5)$ satisfaz a essa equação, visto que:

$$y^2 \pmod p = x^3 + x \pmod p$$

$$(5)^2 \pmod{23} = (9)^3 + (9) \pmod{23}$$

$$25 \pmod{23} = (729 + 9) \pmod{23}$$

$$25 \pmod{23} = 738 \pmod{23}$$

$$2 = 2 \Rightarrow (9, 5) \in E(\mathbb{F}_{23})$$

Nesse exemplo, os 23 pontos que satisfazem a equação são:

(0,0) (1,5) (1,18) **(9,5)** (9,18) (11,10) (11,13) (13,5) (13,18) (15,3) (15,20) (16,8)
(16,15) (17,10) (17,13) (18,10) (18,13) (19,1) (19,22) (20,4) (20,19) (21,6) (21,17)

Esses pontos que, junto com o ponto O, compõem a curva elíptica dada, também podem ser apresentados sob a forma gráfica. Conforme a Figura 3.7, pode ser verificado que o conjunto de pontos, embora não se assemelhe ao formato das curvas apresentadas no plano dos reais, define uma simetria em torno do eixo $y = 11,5$. Esses pontos "simétricos" pelo eixo imaginário, definem todos os pontos e seus respectivos "negativos" na curva (ver Equação 3.7).

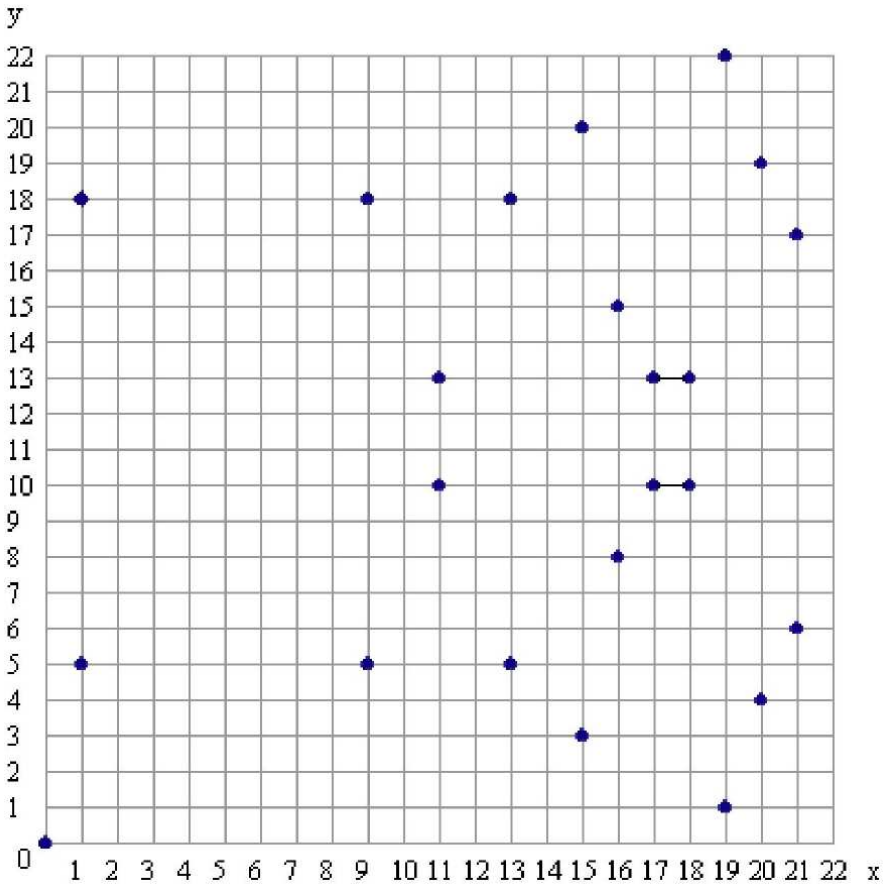

Figura 3.7: Curva elíptica $y^2 = x^3 + x (\text{mod } p)$ em $\mathbb{F}_{23}$

3.2.3 Curvas Elípticas sobre Corpos Finitos de Característica Dois

O conjunto $\mathbb{F}_{2^m}$ é composto por *strings* de m bits e possui dois principais tipos de representação: Representação polinomial e representação de base ótima. Segundo [35], para implementações de *hardware*, a representação de base ótima é a melhor opção, ao passo que em *software*, o uso da representação polinomial é mais indicado. Devido ao caráter mais simplificado, a representação polinomial é a preferida em todas as publicações de cunho tutorial, logo, aqui também, será exemplificado $\mathbb{F}_{2^m}$ através do seu emprego. Em [35] é possível ver, de forma simplificada, o uso da representação de base ótima. Uma terceira representação, chamada representação de subcorpo, também pode ser conferida em [22].

A conveniência do uso de um conjunto de *strings* de m bits está no fato da mesma se aproximar mais da representação "natural" de palavras de dados de m bits em memória [31]. Como resultado do corpo $\mathbb{F}_{2^m}$ ser "de característica 2", a equação que representa uma curva elíptica em $\mathbb{F}_{2^m}$ é ligeiramente diferente das Equações 3.2 e 3.11, apresentadas anteriormente. Vale ressaltar que, essa equação (apresentada abaixo) é aplicável tanto nas representações polinomiais de $\mathbb{F}_{2^m}$ quanto nas de base ótima [34][35], possuindo como única restrição: a, $b \in \mathbb{F}_{2^m}$ e $b \neq 0$ [31][34].

$$y^2 + xy = x^3 + ax^2 + b \in \mathbb{F}_{2^m} \tag{3.15}$$

A representação polinomial de $\mathbb{F}_{2^m}$ é definida por um conjunto de polinômios binários (coeficientes binários) de grau $\leqslant m-1$, conforme definido a seguir [31][34]:

$$\mathbb{F}_{2^m} = \{a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + ... + a_1x + a_0\} \forall a_i \in \{0, 1\} \tag{3.16}$$

Os elementos desse conjunto $\mathbb{F}_{2^m}$ (que contém 2^m elementos) também podem ser escritos na forma de vetor: (a_{m-1}, ..., a_1, a_0). Dessa forma, é possível ter um conjunto $\mathbb{F}_{2^4}$ representado, de forma equivalente, pelos conjuntos abaixo (com $2^4 = 16$ elementos):

$$\mathbb{F}_{2^4} = \{(0001), (0010), (0100), (1000), (0011), (0110), (1100), (1011), (0101), ...\}$$

$$= \{(1), (x), (x^2), (x^3), (x+1), (x^2+x), (x^3+x^2), (x^3+x+1), (x^2+1), (x^3+x), ...\}$$

Assim como em $\mathbb{F}_p$, quando são realizadas operações em elementos desse conjunto, o resultado deve produzir um polinômio também em $\mathbb{F}_{2^m}$, de forma que todas as operações devolvam valores fechados nesse conjunto. Isso é obtido através do uso de um polinômio irredutível de ordem m: $f(x)$ [31][16]. Esse polinômio, também utilizado ao se determinar uma curva elíptica em $\mathbb{F}_{2^m}$, tem o seguinte formato [34]:

$$f(x) = x^m + f_{m-1}x^{m-1} + f_{m-2}x^{m-2} + ... + f_1x + f_0 \tag{3.17}$$

Operações de adição (XOR bit a bit) e multiplicação (módulo $f(x)$) são as mais importantes, pois delas podem ser derivadas subtração e exponenciação (multiplicação múltipla) em $\mathbb{F}_{2^m}$ [34].

Outra definição importante dentro do contexto do uso do $\mathbb{F}_{2^m}$, é o elemento gerador desse conjunto, chamado de g. Um elemento gerador g é um dos elementos de $\mathbb{F}_{2^m}$, a partir do qual é possível gerar todo o conjunto, bastando calcular as potências desse elemento [34]. Dessa forma, no exemplo de $\mathbb{F}_{2^4}$ visto acima foram utilizados o gerador $g = (0010)$ (em forma de polinômio: $\{x\}$) e a função irredutível $f(x) = x^4 + x + 1$. Os demais elementos daquele conjunto foram resultado da potenciação desse valor quando, então, foi obtido:

$\mathbb{F}_{2^4} = \{g^0, g^1, g^2, g^3, g^4, g^5, g^6, g^7, g^8, g^9, g^{10}, g^{11}, g^{12}, g^{13}, g^{14}, g^{15}\}$, onde:
$g^0 = \{1\} = \{0x^3 + 0x^2 + 0x + 1\} = (0001)$
$g^1 = \{x\} = \{0x^3 + 0x^2 + 1x + 0\} = (0010)$
$g^2 = \{x^2\} = \{0x^3 + 1x^2 + 0x + 0\} = (0100)$
$g^3 = \{x^3\} = \{1x^3 + 0x^2 + 0x + 0\} = (1000)$
$g^4 = \{x^4\} \bmod f(x) = \{-x - 1\} = \{0x^3 + 0x^2 - 1x - 1\} = (0011)$
$g^5 = \{x^5\} \bmod f(x) = \{-x^2 - x\} = \{0x^3 - 1x^2 - 1x + 0\} = (0110)$
$g^6 = \{x^6\} \bmod f(x) = \{-x^3 - x^2\} = \{-1x^3 - 1x^2 + 0x + 0\} = (1100)$

$g^7 = \{x^7\} \bmod f(x) = \{-x^3 + x + 1\} = \{-1x^3 + 0x^2 + 1x + 1\} = (1011)$

$g^8 = \{x^8\} \bmod f(x) = \{x^2 + 2x + 1\} = \{0x^3 + 1x^2 + 2x + 1\} = (0101)$

$g^9 = \{x^9\} \bmod f(x) = \{x^3 + 2x^2 + x\} = \{1x^3 + 2x^2 + 1x + 0\} = (1010)$

$g^{10} = \{x^{10}\} \bmod f(x) = \{2x^3 + x^2 - x - 1\} = \{2x^3 + 1x^2 - 1x - 1\} = (0111)$

$g^{11} = \{x^{11}\} \bmod f(x) = \{x^3 - x^2 - 3x - 2\} = \{1x^3 - 1x^2 - 3x - 2\} = (1110)$

$g^{12} = \{x^{12}\} \bmod f(x) = \{-x^3 - 3x^2 - 3x - 1\} = \{-1x^3 - 3x^2 - 3x - 1\} =$
(1111)

$g^{13} = \{x^{13}\} \bmod f(x) = \{-3x^3 - 3x^2 + 1\} = \{-3x^3 - 3x^2 + 0x + 1\} =$
(1101)

$g^{14} = \{x^{14}\} \bmod f(x) = \{-3x^3 + 4x + 3\} = \{-3x^3 + 0x^2 + 4x + 3\} = (1001)$

$g^{15} = \{x^{15}\} \bmod f(x) = \{4x^2 + 6x + 3\} = \{0x^3 + 4x^2 - 6x + 3\} = (0001)$

Em aplicações reais de curvas elípticas sobre $\mathbb{F}_{2^m}$, o valor de m deve ser tal que permita a geração de uma tabela (similar a apresentada acima) grande o suficiente, de forma a tornar o sistema imune a quebras [34]. Hoje em dia, conforme visto na Figura 2.4, o uso de $m = 160$ tem se mostrado uma boa opção.

As equações a seguir, definem as operações de soma de pontos numa curva elíptica sobre $\mathbb{F}_{2^m}$. Observa-se que essas equações [33][31] são ligeiramente diferentes das vistas para $\mathbb{F}_p$ (ver Equações 3.13 e 3.14).

$$P_3(x_3, y_3) = P_1(x_1, x_1) + P_2(x_2, y_2), \text{ onde } P_3 \neq O$$

$$x_3 = (\lambda^2 + \lambda + a + x_1 + x_2) \in \mathbb{F}_{2^m} \tag{3.18}$$

$$y_3 = ((x_1 + x_3)\lambda + x_3 + y_1) \in \mathbb{F}_{2^m} \tag{3.19}$$

onde:

$$x_1 \neq x_2 \Rightarrow \lambda = \frac{(y_2 + y_1)}{(x_2 + x_1)} \in \mathbb{F}_{2^m}$$

$$x_1 = x_2 \Rightarrow \lambda = \frac{(x_1^2 + y_1)}{x_1} \in \mathbb{F}_{2^m}$$

Exemplo de Curvas Elípticas sobre $\mathbb{F}_{2^m}$ ($\mathbb{F}_{2^4}$) [34]:

Considerando agora uma curva elíptica $E(\mathbb{F}_{2^4})$ com a equação $y^2 + xy = x^3 + g^4x^2 + 1$ ($a = g^4$ e $b = g^0 = 1$), pode ser observado que o ponto (g^5, g^3) satisfaz a essa equação, visto que:

$$y^2 + xy = x^3 + g^4x^2 + 1$$

$$(g^3)^2 + (g^5)(g^3) = (g^5)^3 + g^4(g^5)^2 + 1$$

$$g^6 + g^8 = g^{15} + g^{14} + 1$$

$$(1100) + (0101) = (0001) + (1001) + (0001)$$

$$(1001) = (1001) \Rightarrow (g^5, g^3) \in E(\mathbb{F}_{2^4})$$

Nesse exemplo, os 15 pontos que satisfazem a equação são:

$$(1, g^{13})\ (g^3, g^{13})\ (g^5, g^{11})\ (g^6, g^{14})\ (g^9, g^{13})\ (g^{10}, g^8)\ (g^{12}, g^{12})\ (1, g^6)\ (g^3, g^8)\ (g^5, g^3)$$
$$(g^6, g^8)\ (g^9, g^{10})\ (g^{10}, g)\ (g^{12}, 0)\ (0, 1)$$

Esses pontos que, junto com o ponto O, compõem a curva elíptica dada, também podem ser apresentados sob a forma gráfica, conforme a Figura 3.8.

A solução do *ECDLP* com corpos primos ou de característica dois, apresenta, aparentemente, a mesma dificuldade para instâncias que utilizam mesmo tamanho (aproximado) para p ou 2^m. Segundo [10] e [22], não existe estudo matemático que tenha comprovado se o *ECDLP* sobre $\mathbb{F}_{2^m}$ é mais difícil ou fácil de resolver, se comparado ao $\mathbb{F}_p$.

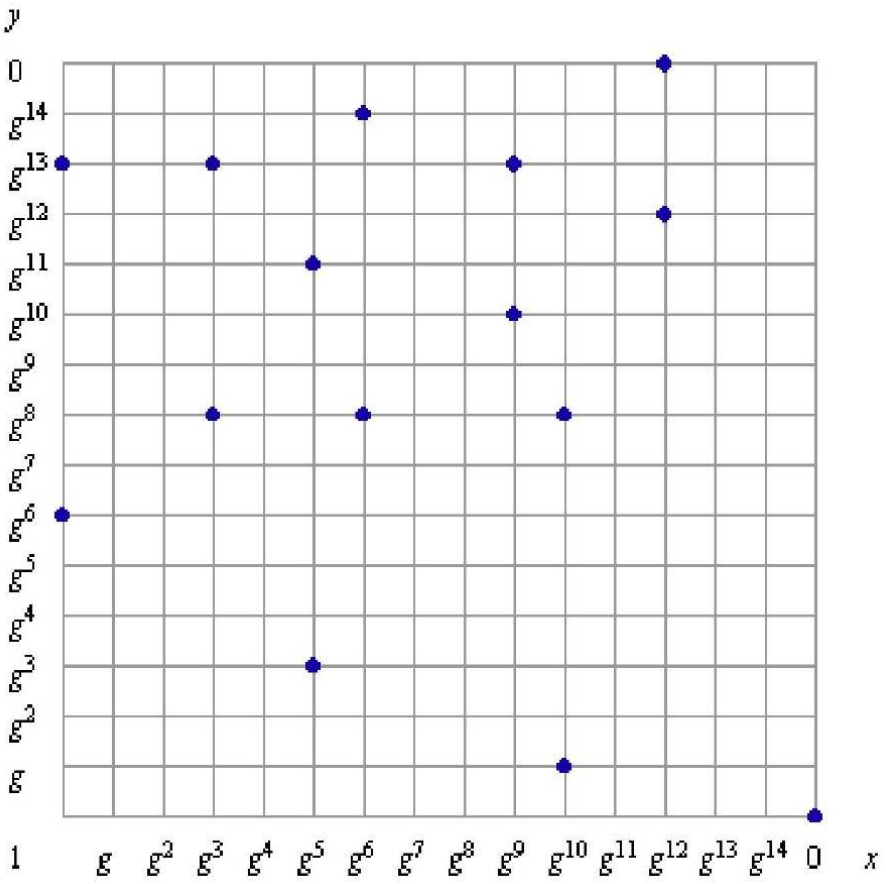

Figura 3.8: Curva elíptica $y^2 + xy = x^3 + g^4x^2 + 1$ em $\mathbb{F}_{2^4}$

3.3 Criptografia com Curvas Elípticas em Aplicações Reais

Apesar das latentes vantagens trazidas pela criptografia com curvas elípticas citadas no Capítulo 2, no contexto de aplicação real de curvas elípticas em criptografia, muitos cuidados devem ser tomados com relação às características e parâmetros gerais escolhidos. Dessa forma, esta seção pretende citar algumas das principais preocupações ao se projetar um sistema baseado em curvas elípticas. No decorrer desta seção, também serão citadas algumas aplicações reais e, ao final, será apresentado um exemplo simples de uma das possíveis aplicações dessa técnica.

3.3.1 Parametrização da Criptografia com Curvas Elípticas

Ao se projetar um sistema de criptografia baseado em curvas elípticas, é necessário determinar, num primeiro momento, quais são as características gerais do sistema, onde serão definidos todos as operações e parâmetros genéricos que todos os componentes irão utilizar. Num segundo momento, cada um dos usuários desse sistema terá que definir seus parâmetros pessoais (chaves), de forma a viabilizar sua participação no sistema.

Ao se determinar as características gerais do sistema, é necessário realizar os seguintes passos [20][22]:

1. Definir a natureza de seu corpo finito $\mathbb{F}_q$ ($\mathbb{F}_p$ ou $\mathbb{F}_{2^m}$), inclusive características do corpo selecionado (valor de p ou polinômio irredutível de ordem m, $f(x)$);
2. Selecionar a representação para os elementos em $\mathbb{F}_q$ (polinomial, base ótima, subcorpos, etc);
3. Implementar aritmética e operações em $\mathbb{F}_q$ (ver Equações 3.13, 3.14, 3.18 e 3.19);
4. Selecionar uma curva apropriada em $\mathbb{F}_q$ (quais parâmetros utilizar para a curva);
5. Definir um ponto gerador em $E(\mathbb{F}_q)$;
6. Definir o mapeamento da mensagem original em pontos de uma curva (*embedding*) - caso o sistema necessite desse mapeamento (exemplo: [21]).

Os três primeiros itens citados acima já foram abordados durante a descrição de curvas elípticas sobre corpos finitos, na Seção 3.2.1.

O quarto item aborda o problema de como melhor escolher os parâmetros da curva elíptica, de forma a tornar seu sistema mais seguro. Na Seção 2.4.1, quando foram discutidos aspectos de segurança e citada a existência de algoritmos específicos para inversão das funções de criptografia, foi observado o fato de que algumas

classes de curva (supersingulares e anômalas) deveriam ser evitadas. Da mesma forma, existem técnicas que possuem o único propósito de auxiliar na escolha de parâmetros de curva apropriados. Entre essas técnicas, pode-se citar: método baseado no teorema de *Hanssen*, o método global, o método da multiplicação de complexos e o método randômico. A descrição desses métodos pode ser obtida em [22]. Discussões mais extensas sobre o que deve ser evitado e quais as "boas práticas" em termos de parametrização dessas curvas também podem ser vistas em [31].

Além da definição da própria curva $E(\mathbb{F}_q)$, como parte dos parâmetros globais que devem ser mantidos públicos, está um ponto denominado ponto "gerador" [36] ou ponto-base $G \in E(\mathbb{F}_q)$ [31]. Esse ponto é uma referência que irá permitir a realização da criptografia. Segundo [31], o ponto G é análogo à base g do problema *DLP* (conforme descrito na Tabela 2.2: determinar x, dados g e $y = (g^x) \mod p$). O ponto G é obtido a partir da escolha de um valor n primo grande tal que $nG = O$ (ponto no infinito) [31][36]. Todos os pontos $P_i \in E(\mathbb{F}_q)$ têm uma "ordem" n_i, tal que, $n_i P_i = O$, dessa forma, valor n é denominado "ordem" de G [31]. Também segundo [31], algo importante, mais nem sempre divulgado, é o fato que n deve ser grande o suficiente de forma a inviabilizar a obtenção de todos os múltiplos de $G : G, 2G, 3G, 4G, ..., (n-1)G$. Mais adiante, será possível visualizar como n e G participam do processo de criptografia.

Um assunto pouco esclarecido nos textos que se propõem a introduzir a técnica de criptografia com curvas elípticas está sendo abordado no item 6, citado anteriormente. Tudo que foi discutido até agora sobre curvas elípticas, tratou somente da base por do trás emprego de curvas elípticas, de suas propriedades, dos pontos dessa curva e da aritmética, operações e natureza desses pontos. Faltava introduzir uma relação clara entre a mensagem original, representada por uma mensagem de texto puro (*plain text*) e os pontos de uma curva elíptica, sobre os quais foram realizadas todas as operações comentadas. Esse procedimento, essencial a alguns tipos de uso de criptografia com curva elíptica, é denominado *message embedding* e consiste em uma forma de mapear a mensagem original (texto puro) em pontos de uma curva elíptica. Isso corresponde a colocar a mensagem original "sobre" [32] a curva elíptica definida. Esses pontos, depois de sofrerem operações parametrizadas pelas chaves

individuais de um usuário, dão origem a um outro conjunto de pontos, que representa os pontos originais "cifrados". Aqui, é observada uma clara diferença entre os métodos de criptografia tradicional e as técnicas baseadas em curvas elípticas: Ao invés de texto cifrado, podem ser transmitidos um conjunto de "pontos cifrados" [17]. Esses "pontos cifrados", ao serem recebidos pelo destinatário, são convertidos, através da chave correspondente, nos pontos originais. Nesse momento, ao se aplicar a rotina de mapeamento invertida, é possível recuperar o texto original e todo o "ciclo criptográfico" se fecha. O mapeamento de caracteres em pontos é citado em [17] e pode ser visto de forma exemplificada em [32].

Os parâmetros do sistema podem sofrer algumas variações, dependendo do tipo da aplicação de criptografia com curvas elípticas em questão. No caso específico da criptografia por identidades (IBE - um dos alicerces da proposta contida neste trabalho), a quantidade de parâmetros é maior, devido à manutenção de pontos globais extras, além de uma espécie de "chave mestra" que fica armazenada dentro de um gerador de chaves privadas, conforme será constatado no Capítulo 4.

Uma vez estabelecidos todos os parâmetros e características gerais nas quais o sistema de criptografia com curvas elípticas deve se basear, basta que cada usuário determine seus parâmetros individuais, ou seja, seu par de chaves pública/privada, e os demais parâmetros locais necessários ao resto da implementação. Dessa forma, cada usuário possui um valor $n_A < n$ (conforme citado anteriormente, n é a ordem do ponto gerador G) como chave privada que possui, como chave pública correspondente, um ponto $P_A = n_A G$. É importante frisar que, geralmente, o ponto P_A é derivado de n_A. Essa derivação nem sempre será verdade pois, no caso do sistema de criptografia por identidades, a chave privada é derivada de uma chave pública escolhida, conforme também será visto no Capítulo 4. Em qualquer situação, tanto P_A quanto G e $P_A \in E(\mathbb{F}_q)$.

O esquema apresentado pela Figura 3.9 assume que a mensagem original M já tenha sido mapeada em um ponto P_M, dentro do grupo finito de pontos de uma curva elíptica $E(\mathbb{F}_q)$. A partir desse momento, quando um usuário A deseja enviar uma mensagem cifrada para B (usando a chave pública P_B), deve realizar o seguinte

procedimento [17][36]:

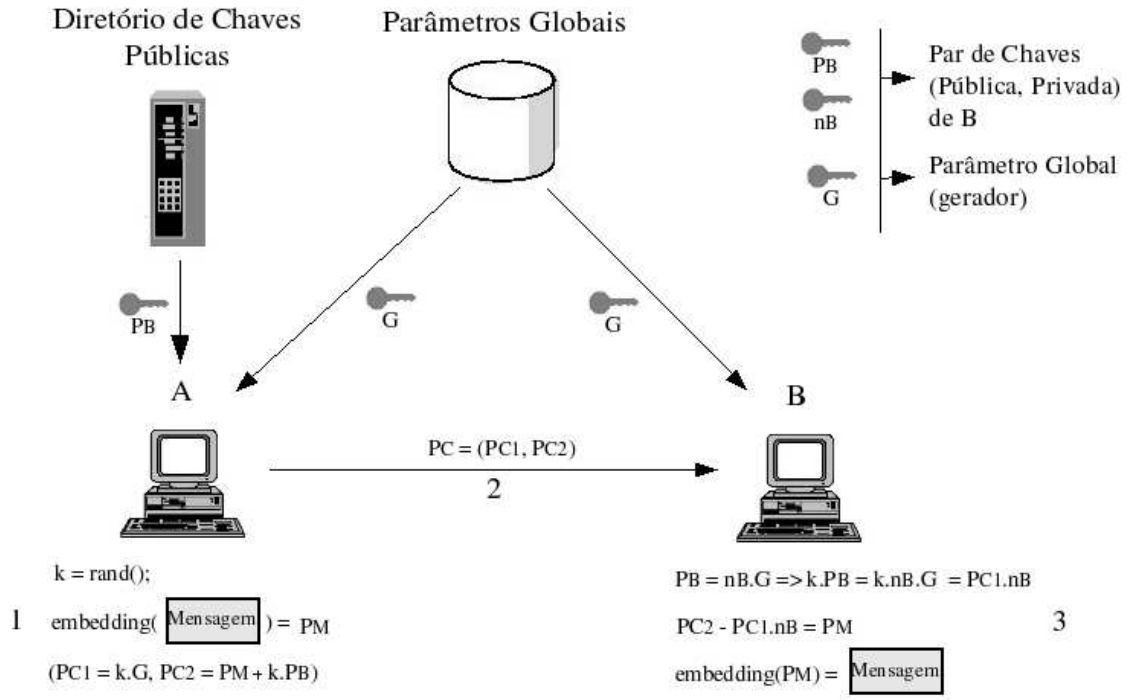

Figura 3.9: Criptografia ElGamal com curva elíptica

Passo 1 A escolhe, aleatoriamente, um inteiro k e calcula, a partir de k, G, P_B e P_M, um par de pontos P_{C1} e P_{C2}: $P_{C1} = (kG)$ e $P_{C2} = (P_M + kP_B)$

Passo 2 A transmite para B o par de pontos cifrados P_C: $P_C = [P_{C1}, P_{C2}]$

Do outro lado, quando B recebe a mensagem (par de pontos P_C), recupera P_M, a partir do segundo ponto (P_{C2}), da seguinte forma:

Passo 3 Embora B não conheça k, sabendo que: $kP_B = kn_BG$ (pois $P_B = n_BG$): $P_{C1}n_B = (kG)n_B = kP_B \Rightarrow kP_B$ "sai" de P_{C1}

Passo 4 Para extrair P_M de P_{C2}, basta calcular $P_{C2} - kP_B \Rightarrow P_{C2} - P_{C1}n_B = (P_M + kP_B) - kP_B = (P_M + [(kG)n_B]) - [(kG)n_B]) = P_M$

A Figura 3.9, bem como o exemplo a seguir, são ilustrações do denominado esquema *ElGamal* para criptografia com chave pública [21]. Esquemas similares para obtenção de texto cifrado, assim como assinatura digital e diversos outros

casos reais de aplicações de curvas elípticas em criptografia, também podem ser encontrados em [33], [17], [22], [31] e [36].

3.3.2 Exemplo de Criptografia com Curvas Elípticas [36]

Considerando uma curva elíptica $E(\mathbb{F}_{751})$ com a equação $(y^2) = (x^3 - x + 188) \mod 751$ ($a = -1$ e $b = 188$), e ponto gerador $G = (0, 376)$. Conseqüentemente, os múltiplos kG do ponto gerador G são ($1 \leqslant k \leqslant 751$):

$G = (0, 376)$
$2G = (1, 376)$
$3G = (750, 375)$
$4G = (2, 373)$
$5G = (188, 657)$
$6G = (6, 390)$
$7G = (667, 571)$
$8G = (121, 39)$
$9G = (582, 736)$
$10G = (57, 332)$
...
$761G = (565, 312)$
$762G = (328, 569)$
$763G = (677, 185)$
$764G = (196, 681)$
$765G = (417, 320)$
$766G = (3, 370)$
$767G = (1, 377)$
$768G = (0, 375)$
$769G = O$ (ponto no infinito)

Quando A decide mandar uma mensagem M para B, ele deve saber, a partir de alguma estratégia [17], a chave pública do destinatário. Se o destinatário desse

exemplo escolheu o valor $n_B = 85$ como chave privada, então a chave pública correspondente à mesma é o ponto $P_B = n_B G = 85(0, 376) \Rightarrow P_B = (671, 558)$. O procedimento de criptografia de P_M passa, então, pelas etapas:

A: Mapeia a mensagem M no ponto $P_M = (433, 253) \in E(\mathbb{F}_q)$

A: Escolhe, digamos, $k = 113$

A: Calcula o par de pontos P_C :

$PC = [(kG), (P_M + kP_B)]$

$PC = [113(0, 376), (443, 253) + 113(671, 558)]$

$PC = [(34, 633), (443, 253) + (47, 416)]$

$PC = [(34, 633), (217, 606)]$

A: Transmite para B o par de pontos cifrados $P_C = [(34, 633), (217, 606)]$

B: Recebe o par de pontos cifrados $P_C = [(34, 633), (217, 606)]$

B: Calcula, a partir de sua chave privada $n_B = 85$:

$(P_M + kP_B) - [n_B(kG)] = (217, 606) - [85(34, 633)]$

$(P_M + kP_B) - [n_B(kG)] = (217, 606) - [(47, 416)]$

$(P_M + kP_B) - [n_B(kG)] = (217, 606) + [(47, -416)]$ (ver Equação 3.7)

$(P_M + kP_B) - [n_B(kG)] = (217, 606) + [(47, 335)]$ $(-416 = 335 \mod 751)$

$(P_M + kP_B) - [n_B(kG)] = (443, 253)$

B: Mapeia ponto $P_M = (433, 253)$ novamente na mensagem M

Embora o exemplo acima seja uma ilustração bastante simplória da aplicação real de curvas elípticas em criptografia, é possível visualizar, através do mesmo, como pontos e operações em grupos finitos desses pontos podem ser utilizadas em processos de criptografia. O próximo capítulo, contudo, pretende abordar de forma mais específica a aplicação de curvas elípticas em criptografia que motivou a principal proposta deste trabalho: A criptografia por identidades, ou IBE, cuja implementação é viável graças a um emprego ainda mais matematicamente sofisticado dessas técnicas.

Capítulo 4

Criptografia Baseada em Identidades: *Identity Based Encryption* (IBE)

A proposta de uso de criptografia por identidades feita por Adi Shamir em 1984 [2], idealizava o uso de identificadores pessoais (que distinguem unicamente um elemento em um grupo) na geração de um par de chaves não-aleatórias para um sistema de criptografia com chaves assimétricas. Com esse par de chaves, seria aplicado um algoritmo criptográfico cujo texto cifrado poderia ser produzido e recuperado com uma e outra chave, da mesma forma como nos sistemas tradicionais. A idéia de poder produzir uma chave pública de forma não-aleatória e de deduzir uma chave privada a partir dela, naquele momento, parecia tão inovadora quanto enigmática sob o ponto de vista de implementação.

Contudo, quase duas décadas mais tarde, uma nova e poderosa aplicação de criptografia com curvas elípticas acabaria surgindo. Assim, em 2001, uma proposta de implementação do modelo sugerido por Shamir pôde viabilizá-lo em sua plenitude, graças ao trabalho de Dan Boneth e Matthew Franklin [3]. O trabalho de Boneth e Franklin visualizou o emprego de curvas elípticas na criação dos sistemas de criptografia por identidades, constituindo a base do modelo que pôde trazer a idéia de [2] ao estágio de uma aplicação real em todas as suas formas.

De fato, conforme será constatado neste capítulo, teoria por trás da IBE acena

como uma potencial melhoria e simplificação do processo de envio de mensagens sigilosas com aplicação de chaves públicas. Conforme mencionado no Capítulo 1, essa melhoria deve-se ao simples fato de que, nesse modelo, essas chaves são obtidas através de identidades "escolhidas" e não por intermédio de uma seqüência de bits gerados aleatoriamente, como acontece nos sistemas de chave pública "tradicionais".

Este capítulo busca detalhar um pouco mais a teoria envolvendo criptografia por identidades, citando alguns dos conceitos básicos, porém necessários, para um entendimento inicial dessa técnica. Serão vistas, também, possíveis aplicações desse tema nas mais diversas áreas de segurança, como sigilo, autenticidade (assinatura digital), troca de chaves, processos de autenticação e controle de integridade, assim como a adoção da IBE dentro do contexto de *e-mail* seguro, aplicação central da proposta contida neste trabalho.

4.1 Teoria Básica

4.1.1 Infra-estrutura de Chave Pública

No Capítulo 2, ao definir sistemas de chave pública (ilustrados através da Figura 2.2), muitos detalhes relativos à geração, obtenção e gerenciamento de chaves não foram comentados. Esses aspectos, de suma importância quando deseja-se implantar esse tipo de tecnologia em um ambiente comercial, são abordados pelo que pode ser denominado de Infra-estrutura de Chave Pública (ICP), também chamada de *Public Key Infraestructure - PKI*.

Infra-estrutura de chave pública foi introduzida no mercado ainda nos anos 80, trazendo alguns conceitos fundamentais ao emprego de criptografia assimétrica no mundo real. Dessa forma, aspectos relacionados a políticas de acesso, tais como: quais usuários têm direito de acesso, em qual rede, em que nível ou por quanto tempo esse acesso está vigente, etc, são unidos à identidade pública de um dos participantes do sistema e, junto com sua chave pública, dão origem ao que pode ser definido como "certificado digital".

Dessa forma, o certificado digital contém, além da identificação pessoal do elemento, sua chave pública e demais informações necessárias ao se realizar uma operação segura, como data de expiração. Esse pacote de dados é "assinado" digitalmente por uma entidade denominada de Autoridade Certificadora, através da chave privada que somente essa autoridade possui. Assim, sempre que se desejar verificar um certificado assinado por uma autoridade certificadora, basta aplicar a chave pública dessa autoridade e constatar que somente ela poderia ter emitido essa espécie de documento eletrônico. Os certificados são, em última instância, uma forma de autenticar e "ligar" a identidade de um elemento à sua chave pública, de forma indissolúvel, por tempo limitado e com características específicas, caso essas sejam necessárias.

A Figura 4.1 é um detalhamento do que já havia sido apresentado na Figura 2.2, incluindo os conceitos de ICP mencionados neste capítulo.

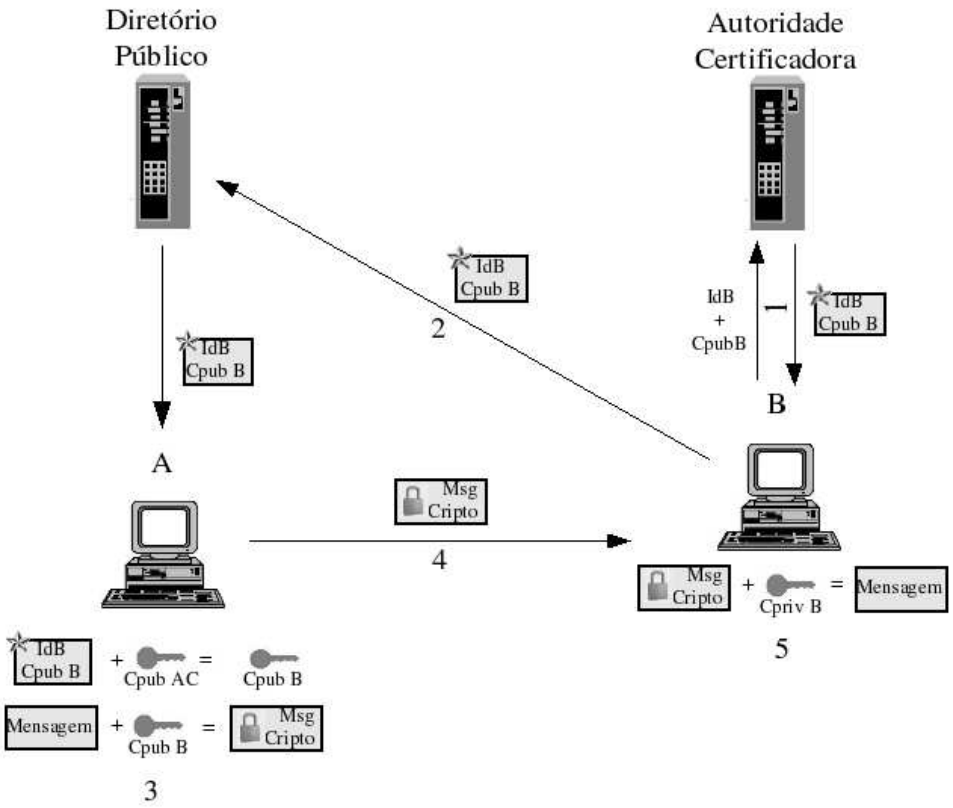

Figura 4.1: Criptografia Assimétrica com Certificação

Passo 1 B solicita um certificado a uma Autoridade Certificadora, passando a sua identidade (IdB) e chave pública ($CpubB$);

Passo 2 B divulga seu certificado em algum diretório de certificados;

Passo 3 A obtém o certificado de B, com o identificador IdB e, aplicando a chave pública da Autoridade Certificadora, verifica a veracidade e validade desse certificado;

Passo 4 A criptografa o texto aplicando $CpubB$ e envia essa mensagem para B, tendo certeza que somente o "verdadeiro" B poderá abrir essa mensagem;

Passo 5 B recupera o texto original aplicando $CprivB$, que somente ele tem.

A aplicação de certificação digital em processos de autenticação é de grande valia quando se deseja saber se o elemento com quem se está conectado é realmente quem se acredita ser. A Figura 4.2 exemplifica como um processo de autenticação pode ser feito a partir da obtenção de certificados entre as partes envolvidas.

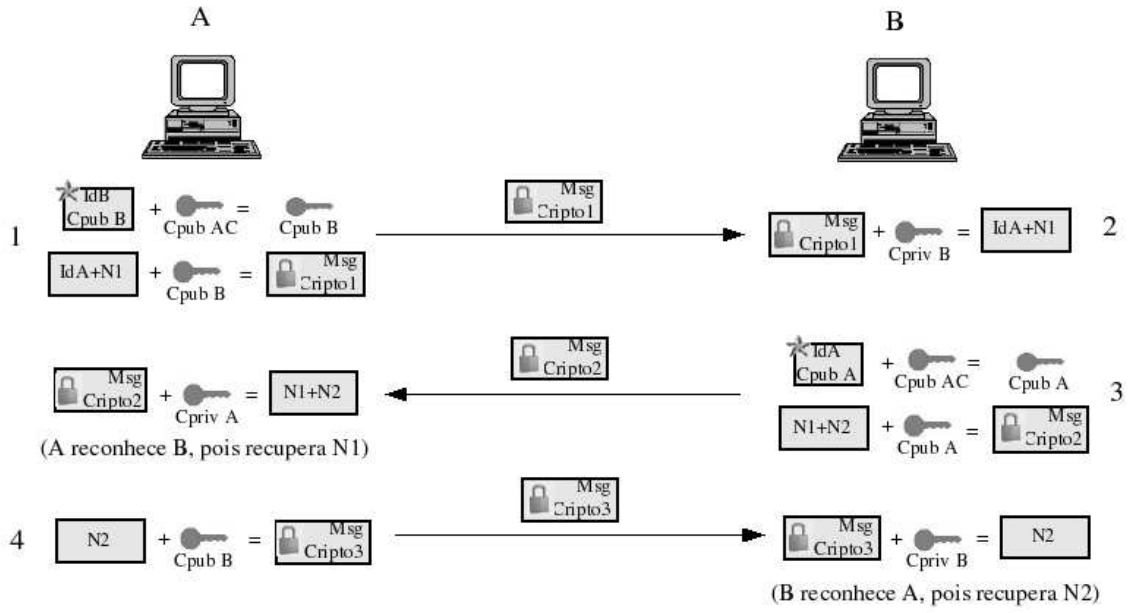

Figura 4.2: Autenticação com Certificação

Passo 1 A verifica o certificado digital de B e utiliza a chave pública contida nesse certificado para enviar a B seu identificador, assim como

um valor $N1$ aleatoriamente escolhido por ele;

Passo 2 B recupera os valores de identificação de A (IdA) e o número gerado por A ($N1$);

Passo 3 B verifica o certificado de A e, com a chave obtida desse certificado, criptografa o valor obtido da mensagem de A ($N1$) através da aplicação de sua chave privada $CprivB$, incluindo outro valor aleatório $N2$;

Passo 4 A utiliza a chave pública de B ($CpubB$) aplicando-a ao valor $N2$ proposto por B.

Pode ser observado o papel fundamental exercido pelas autoridades certificadoras, uma vez que elas são a base em que o sistema tem que se apoiar para funcionar corretamente. Um vazamento de informação que deixe a chave privada da autoridade vulnerável à consultas ou uma técnica de criptografia por chave pública que não seja segura (conforme visto na Seção 2.4.1) podem comprometer todo o funcionamento desse sistema. Prevendo essas possibilidades, algumas situações de contorno, como revogação (emitindo as denominadas listas de revogação de certificado) e redivulgação de certificados, podem ser necessárias de forma a reestabilizar o sistema.

A crescente expansão de sistemas baseados em certificação digital acaba inviabilizando, na prática, a concentração em uma única autoridade a certificação de toda a comunidade presente nesse sistema. Dessa forma, estruturas hierárquicas, denominadas de modelos de certificação, podem ser necessárias de forma a criar uma "rede" hierarquizada de autoridades certificadoras, a fim de suprir a grande demanda criada por sistemas de maior porte.

Os exemplos de sistemas de criptografia assimétrica representados pelas Figuras 4.1 e 4.2 são totalmente dependentes da existência da autoridade certificadora, que funciona como uma indispesável terceira parte na qual todos confiam. Dessa forma, essa autoridade (AC) é o "elo" de segurança existente entre as partes A e B, visto que se A confia em AC e B também, sempre que AC ratificar uma ação de A para B ou de B para A, ambos irão confiar na segurança do processo. Contudo, essa dependência traz alguns inconvenientes, conforme observa-se a seguir:

- **Dificuldade de localização dos certificados.** A ausência de um diretório-padrão pode fazer com que a localização de certificados seja difícil, senão impossível, independente de as principais partes envolvidas em uma transação estarem ou não conectadas e disponíveis de forma simultânea.

- **Necessidade de cadastramento prévio de usuários.** Antes de enviar ou receber mensagens e transações seguras, o usuário tem que se fazer conhecido pelo sistema ICP. Como exemplo dessa necessidade, clientes que quiserem conectar-se de forma segura com um servidor *Web* que não é conhecido da autoridade certificadora, simplesmente não conseguirão fazê-lo.

- **Dificuldade de Gerenciamento do Sistema de Chaves Públicas.** Conforme já foi adiantado neste capítulo (ao se introduzir modelos de certificação), o crescimento da comunidade de usuários e a conseqüente explosão da demanda por esse tipo de serviço traz problemas de gerenciamento, além de dificuldades associadas aos tempos para revogação de certificados que, eventualmente, deixem de ser válidos (publicação de listas de revogação X estruturas hierárquicas extensas).

Os fatores acima motivam a pesquisa e busca de soluções que consigam simplificar o cenário de criptografia com chave pública. Novos cenários onde, idealmente, poderia se dispensar a figura de uma autoridade certificadora (bem como os certificados) sem, contudo, comprometer sua eficiência e segurança do sistema, seriam muito bem vindos nesse caso. É exatamente com o intuito de ocupar esse espaço que surge a criptografia de chave pública por identidades, que será apresentada na próxima seção.

4.1.2 Criptografia Assimétrica por Identidades

A idéia de projetar um sistema de criptografia com chave pública por identidades (*Identity Based Encryption - IBE*), conforme citado no início deste capítulo, surgiu em 1984, quando Adi Shamir divulgou sua idéia [2] de um novo sistema baseado em

chaves assimétricas. Nesse novo sistema, os elementos seriam capazes de comunicar-se de forma segura ou de verificar assinaturas, sem a prévia necessidade de troca de chave pública ou obtenção de uma chave privada correspondente. Ainda segundo Shamir, seu esquema assumia a existência de um elemento confiável, responsável pela geração de chaves privadas, a partir de chaves públicas escolhidas. Esse elemento pode ser denominado como Gerador de Chaves Privadas - GCP (*Private Key Generator - PKG*).

De acordo com a proposta [2], esses sistemas seriam ideais para grupos fechados de usuários, como executivos de multinacionais ou gerentes das agências de grandes bancos. Contudo, ele também se aplicaria perfeitamente mesmo quando utilizado por redes de larga escala, com centenas de geradores de chave privadas e milhões de usuários.

O esquema de criptografia por identidades seria, portanto, um sistema de criptografia com chave pública com uma nova particularidade: Ao invés da tradicional geração aleatória de um par de chaves (pública e privada) e subseqüente publicação de uma dessas chaves (a pública), um usuário poderia escolher sua chave pública a partir, por exemplo, da concatenação de seu nome com o nome da sua empresa. Dessa forma, qualquer combinação de nomes, números, registros cadastrais, endereços postais e até datas, podem ser utilizados em conjunto na geração de uma chave pública única para um determinado participante desse sistema. Qualquer outro usuário que deseje, conseqüentemente, enviar uma mensagem sigilosa destinada a essa identificação que, na prática, acaba se convertendo na própria chave pública do destinatário, poderia fazê-lo, sem a necessidade de localização da mesma de diretórios públicos de chaves e/ou certificados, como nos esquemas vistos até agora. Verificações de mensagens assinadas por chaves privadas que somente poderiam ser obtidas através dessa identificação, também não teriam sua autoria negada, pois somente o GCP (entidade confiável) ou o próprio gerador da assinatura têm acesso à mesma.

Nesse ponto, pode-se observar que o gerador de chaves privadas é o ponto central de funcionamento dessa estrutura, possuindo uma grande carga de responsabilidade

dentro dela. Assim como vazamentos de informações e senhas poderiam afetar a segurança das autoridades certificadoras, o GCP também pode, a partir de sua violação, comprometer a estabilidade dos sistemas baseados em identidade.

Conseqüentemente, a segurança de sistemas baseados em identidades depende de [2]:

- Segurança dos algoritmos de criptografia utilizados na sua implementação (comum a todos sistemas assimétricos).
- Proteção das informações secretas armazenadas dentro do gerador de chaves privadas.
- Rigidez na verificação e checagem da real identidade de um participante antes da geração e fornecimento de sua chave privada.
- Cuidados dos usuários ao armazenarem suas chaves privadas (também comum a todos sistemas assimétricos).

Situações de contorno aplicáveis ao vazamento de informação ou violações dos princípios citados acima, assim como nos ambientes certificados, também podem ser necessárias. Assim, a validade de chaves públicas também pode fazer parte da geração dos próprios identificadores que compõem uma chave pública, de forma similar às datas de validades presentes em certificados digitais. Sugestões como essas estão presentes no trabalho de [3], que será melhor abordado na Seção 4.1.3.

A Figura 4.3, criada a partir de [2], mostra como é possível comparar os três sistemas - criptografia por chave privada, criptografia por chave pública (tradicionais) e por identidades - na transmissão de informações sigilosas.

Também criada a partir de [2], a Figura 4.4 mostra a comparação entre sistemas de criptografia por chave pública e por identidades em esquemas de assinatura digital.

A proposta de Shamir, contudo, na época em que foi idealizada, ainda não permitia sua integral implementação devido à limitações relacionadas às tecnologias

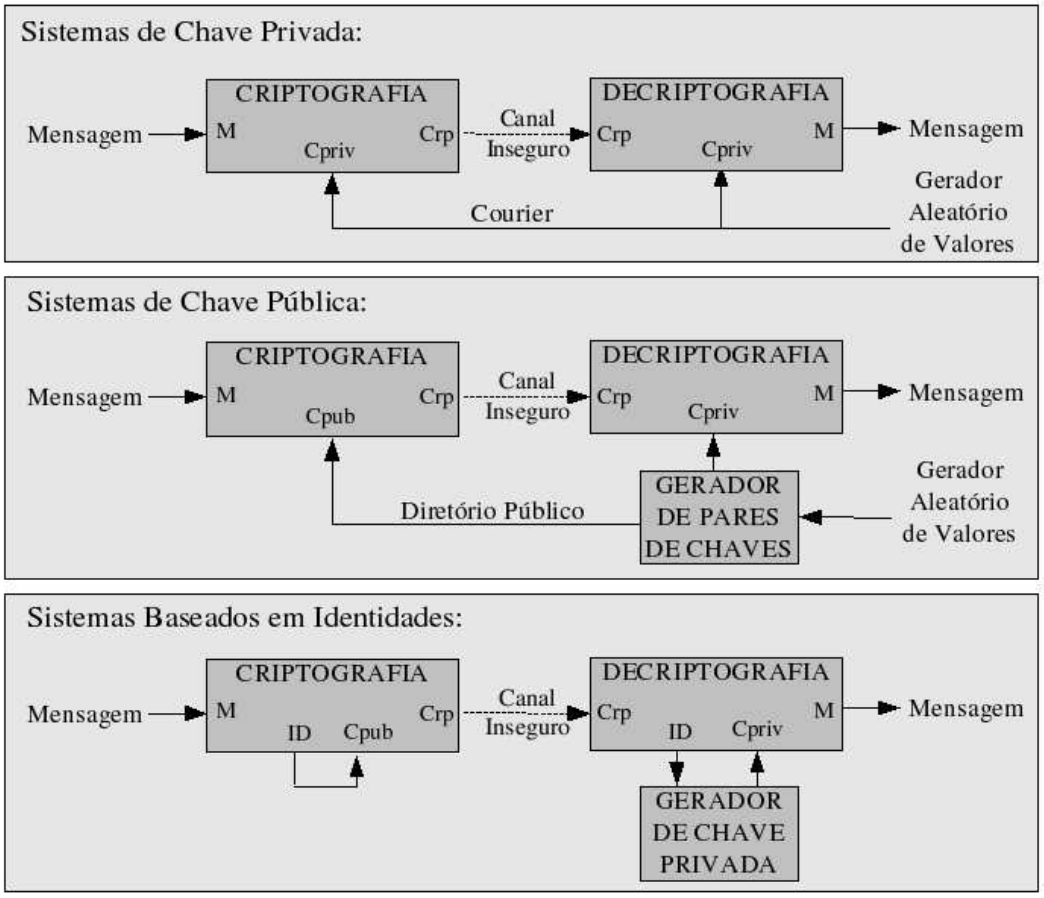

Figura 4.3: Criptografia por Chave Privada, por Chave Pública e por Identidades

e algoritmos de chave pública disponíveis até então. Apesar disso, em [2] é possível encontrar sugestões de implementação de esquemas de assinatura digital baseados em identidades, utilizando como ferramenta o RSA. Segundo Shamir, infelizmente, o RSA não possuía todas as propriedades necessárias ao funcionamento pleno de sua proposta, o que impossibilitou-o de apresentar sugestões de implementações integrais para a sua idéia. Na próxima seção será melhor abordado como operações com pontos em curvas elípticas possibilitaram que esse objetivo pudesse, finalmente, ser alcançado.

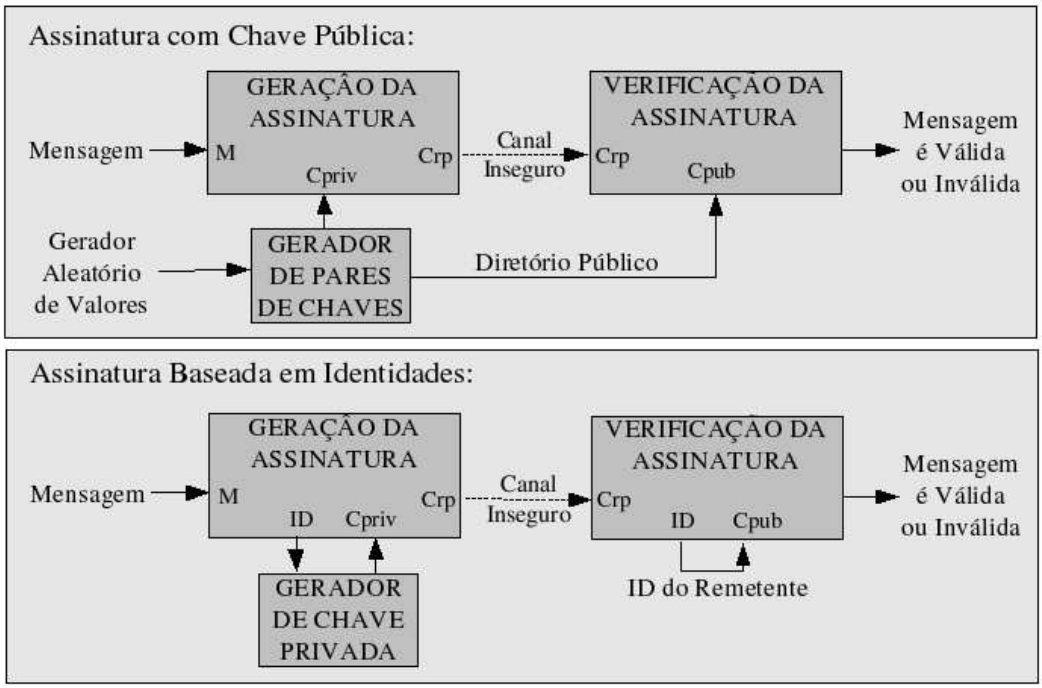

Figura 4.4: Assinatura por Chave Pública e por Identidades

4.1.3 Curvas Elípticas Aplicadas à IBE

Desde sua descrição inicial [2], alguns trabalhos e tentativas de plena implementação de criptografia por identidades vinham sendo realizados. Trabalhos como [37], [38], [39], [40] e [41] podem ser citados como referências a esses esforços. Algumas outras soluções de assinatura e autenticação, alternativas à proposta de [2], também foram levantadas nos trabalhos [42] e [43].

Finalmente, o trabalho [3] traz uma proposta completa para criptografia por identidades. O contexto dessa proposta contém algumas definições importantes, que serão citadas no decorrer desta seção. Contudo, por tratar-se de um assunto notoriamente extenso e complexo, esse texto irá se concentrar nos principais aspectos necessários ao entendimento dessa técnica. Dessa forma, referências serão citadas como valioso complemento ao conteúdo que será apresentado aqui.

Um dos mais importantes fatores para a bem-sucedida proposta de [3], está na utilização de curvas elípticas em conjunto com os chamados mapas bilineares,

conforme definidos a seguir.

Seja $\mathbb{G}_1$ e $\mathbb{G}_2$ dois grupos de ordem q para algum valor primo grande q. A alternativa proposta em [3], utiliza um mapa bilinear $\hat{e}$: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ entre esses dois grupos. Conforme citado em [3], esse mapa tem que satisfazer às seguintes propriedades:

1. Bilinearidade: Um mapeamento $\hat{e}$: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ é dito **bilinear** se:

$$\hat{e}(aP,\, bQ) = \hat{e}(P,\, Q)^{ab} \;\forall\; P,\, Q \in \mathbb{G}_1\,,\; \forall\; a,\, b \in \mathbb{Z}$$

2. O mapa não pode ser degenerativo, ou seja, ele não mapeará todos os pares em $\mathbb{G}_1 \times \mathbb{G}_1$ para o elemento identidade do grupo $\mathbb{G}_2$. E mais, como $\mathbb{G}_1$ e $\mathbb{G}_2$ são grupos de ordem prima, isso implica que, se P é um gerador em $\mathbb{G}_1$, logo $\hat{e}(P,\, P)$ é um gerador em $\mathbb{G}_2$ [1].

3. O mapa deve ser computacionalmente viável, sendo necessário um algoritmo eficiente para calcular $\hat{e}(P,\, Q)$, para qualquer $P,\, Q \in \mathbb{G}_1$

Aplicando essas propriedades, é possível alcançar a propriedade de simetria entre pares, ou seja:

$$\begin{aligned}
\hat{e}(S,\, T) &= \hat{e}(aP,\, bP) \;\forall\; a,\, b \in \mathbb{Z} \\
&= \hat{e}(P,\, P)^{ab} \\
&= \hat{e}(bP,\, aP) \\
&= \hat{e}(T,\, S)
\end{aligned}$$

Conseqüentemente:

$$\hat{e}(S,\, T) = \hat{e}(T,\, S),\; \forall\; S,\, T \in \mathbb{G}_1$$

e

$$\hat{e}(aP,\, bP) = \hat{e}(bP,\, aP) \;\forall\; a,\, b \in \mathbb{Z}$$

[1] Se g é gerador de $\mathbb{F}_q$, logo $g^j \bmod q$ apresenta $\mathbb{F}_q$ como imagem $\forall\; j,\; 0 \leqslant j < p$.

Uma vez satisfazendo às propriedades mencionadas anteriormente, um mapeamento é dito mapeamento "admissível".

A proposta de [3] sugere, também, a aplicação desses grupos $\mathbb{G}_1$ e $\mathbb{G}_2$ ao universo de pontos de uma curva elíptica. Assim, segundo Boneth e Franklin, tem-se que $\mathbb{G}_1$ é um subgrupo de pontos "aditivos" dentro da curva $E(\mathbb{F}_q)$, enquanto $\mathbb{G}_2$ é o subgrupo de um grupo "multiplicativo" de um corpo finito $\mathbb{F}^*_{p^2}$, no qual pares de pontos são "mapeados" em valores. O trabalho de [3] apresenta o emparelhamento de *Weil* como opção de mapeamento "admissível" dentro do contexto de curvas elípticas.

Existe, contudo, uma outra variação de emparelhamento em curvas elípticas que também gera um mapeamento "admissível", denominado emparelhamento de *Tate*. Em linhas gerais, esse emparelhamento utiliza dois grupos "aditivos" com pontos de curvas elípticas aplicadas a diferentes corpos finitos: $E(\mathbb{F}_q)$ e $E(\mathbb{F}_{q^k})$ ($k \in \mathbb{Z}^*$ é o "grau de imersão" da curva considerada [44]) - $\mathbb{G}_0$ e $\mathbb{G}_1$, respectivamente - que são mapeados em um grupo "multiplicativo" $\mathbb{G}_2$ ($\mathbb{F}^*_{q^k}$). Assim, a definição desses mapas bilineares ficaria ligeiramente modificada:

$$\hat{e}\colon \mathbb{G}_0 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$$

Considerada ainda mais eficiente em tempo de execução, o emparelhamento de *Tate* está presente nos trabalhos [44], [45] e [46]. Mais recentemente, a dissertação de [47], que se propõe a apresentar sistemas criptográficos baseados em identidades pessoais, também descreve de forma mais específica esse emparelhamento. Cabe ressaltar que a biblioteca IBE [48], utilizada na implementação deste trabalho, é mais um exemplo de adoção do emparelhamento de *Tate*. Após uma análise no histórico de desenvolvimento de [48], pôde-se observar que essa técnica também apresentou um melhor desempenho nessa biblioteca, após experiências iniciais com versões anteriores implementando *Weil*.

Ao definir criptografia por identidades, os autores da proposta [3] apresentaram um esquema composto por quatro algoritmos básicos: **Configura**, **Extrai**, **Criptografa** e **Decriptografa**, conforme descritos a seguir. Vale observar que, para efeito de simplificação, serão mantidas aqui as definições de mapas bilineares assim

como definidas no trabalho de [3], ou seja, considerando apenas um único subgrupo "aditivo" $\mathbb{G}_1$ como entrada do mapeamento.

Configura: A partir de uma entrada $k \in \mathbb{Z}^+$, retorna os parâmetros do sistema (*params*) e uma chave-mestra (s). A chave-mestra é fundamental no trabalho do GCP (conforme definido na Seção 4.1.2), pois permite ao mesmo gerar e fornecer a chave privada de qualquer elemento do sistema, a partir da sua identidade. Essa parametrização também inclui a descrição de um espaço finito de mensagens $\mathcal{M}$, assim como a descrição de um espaço finito de cifras $\mathcal{C}$. Cabe ressaltar que todos esses parâmetros serão públicos ao sistema, com exceção de s que será secreta e interna ao GCP. O texto de [3] apresenta esse algoritmo como sendo composto pelas seguintes fases:

1. A partir de k, gerar um primo q, dois grupos $\mathbb{G}_1$ e $\mathbb{G}_2$ de ordem q e um mapa bilinear $\hat{e}$: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$. Escolher, aleatoriamente, um gerador $P \in \mathbb{G}_1$.
2. Gerar, também aleatoriamente, $s \in \mathbb{Z}_q^*$ e calcular $P_{pub} = sP$ [2].
3. Escolher uma função de *hash* H_1: $\{0, 1\}^* \longrightarrow \mathbb{G}_1$. Escolher outra função de *hash* H_2: $\{0, 1\}^n$ para algum n. O espaço de mensagens é, então, $\mathcal{M} = \{0, 1\}^n$, enquanto o espaço de cifras é $\mathcal{C} = \mathbb{G}_1 \times \{0, 1\}^n$. Os parâmetros públicos do sistema são $params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, H_1, H_2 \rangle$ e a chave mestra, $s \in \mathbb{Z}_q^*$.

Extrai: Tem como entrada *params*, s e um ID escolhido na forma $ID \in \{0, 1\}^*$ e como saída a chave privada d_{ID}. Assim, ID corresponde a uma chave pública previamente escolhida e d_{ID} será sua chave privada correspondente, fornecida por esse algoritmo. Suas fases são:

1. Calcular $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$
2. Obter $d_{ID} = sQ_{ID}$

Criptografa: Tem como entrada *params*, ID e $M \in \mathcal{M}$ e como saída o texto cifrado $C \in \mathcal{C}$. Suas fases são:

[2] Aplicação de álgebra elíptica do Capítulo 3.

1. Calcular $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$
2. Gerar, aleatoriamente, $r \in \mathbb{Z}_q^*$
3. Obter o texto cifrado $C = \langle rP,\, M \oplus H_2(g_{ID}^r) \rangle$, onde $g_{ID} = \hat{e}(Q_{ID},\, P_{pub}) \in \mathbb{G}_2^*$.

Decriptografa: Tem como entrada $params$, d_{ID} e $C \in \mathcal{C}$ e como saída o texto cifrado $M \in \mathcal{M}$. Considerando $C = \langle U,\, V \rangle \in \mathcal{C}$ o texto cifrado gerado pelo algoritmo **Criptografa** com a identidade ID, para decriptografar C utilizando sua chave privada $d_{ID} \in \mathbb{G}_1^*$, faz-se: $V \oplus H_2(\hat{e}(d_{ID},\, U)) = M$.

Observando esses algoritmos é possível notar que, durante **Criptografa**, M é criptografado (via operação OU-EXCLUSIVO) com o *hash* de g_{ID}^r. Durante **Decriptografa**, V é decriptografado (também via OU-EXCLUSIVO) com o *hash* de $\hat{e}(d_{ID},\, U)$. Isso funciona, pois, ao se aplicar as propriedades de mapas bilineares, obtem-se que $g_{ID}^r = \hat{e}(d_{ID},\, U)$:

$$\hat{e}(d_{ID},\, U) = \hat{e}(sQ_{ID},\, rP) = \hat{e}(Q_{ID},\, P)^{sr} = \hat{e}(Q_{ID},\, P_{PUB})^r = g_{ID}^r$$

Ao se colocar em uso os algoritmos citados, tem-se que:

$$\textbf{Configura}(k) = (params,\, s,\, \mathcal{M},\, \mathcal{C})$$

$$\textbf{Extrai}(params,\, s,\, ID) = d_{ID}$$

e, finalmente, $\forall\, M \in \mathcal{M}$:

$$C = \textbf{Criptografa}(params,\, ID,\, M)$$

$$\textbf{Decriptografa}(params,\, C,\, d_{ID}) = M$$

A Figura 4.5 mostra como, utilizando cada uma dessas definições, é possível enviar uma mensagem criptografada entre os elementos A e B, participantes de um sistema baseado em identidades. Através dessa figura, é possível também observar como aplicar as propriedades de mapeamentos "admissíveis" na viabilização do envio

e recebimento de mensagens sigilosas. Conforme sugerido nos algoritmos básicos **Criptografa** e **Decriptografa**, esse exemplo também aplica a operação $\oplus$ para realizar a criptografia simétrica através da chave k.

Ainda a respeito da Figura 4.5, cabe ressaltar que a operação $\oplus$ não necessariamente será utilizada (principalmente num cenário de aplicação real) na criptografia simétrica da mensagem através da chave k. Como exemplo disso, na implementação proposta neste trabalho, essa tarefa é feita com o uso de uma implementação do DES (Seção 2.1).

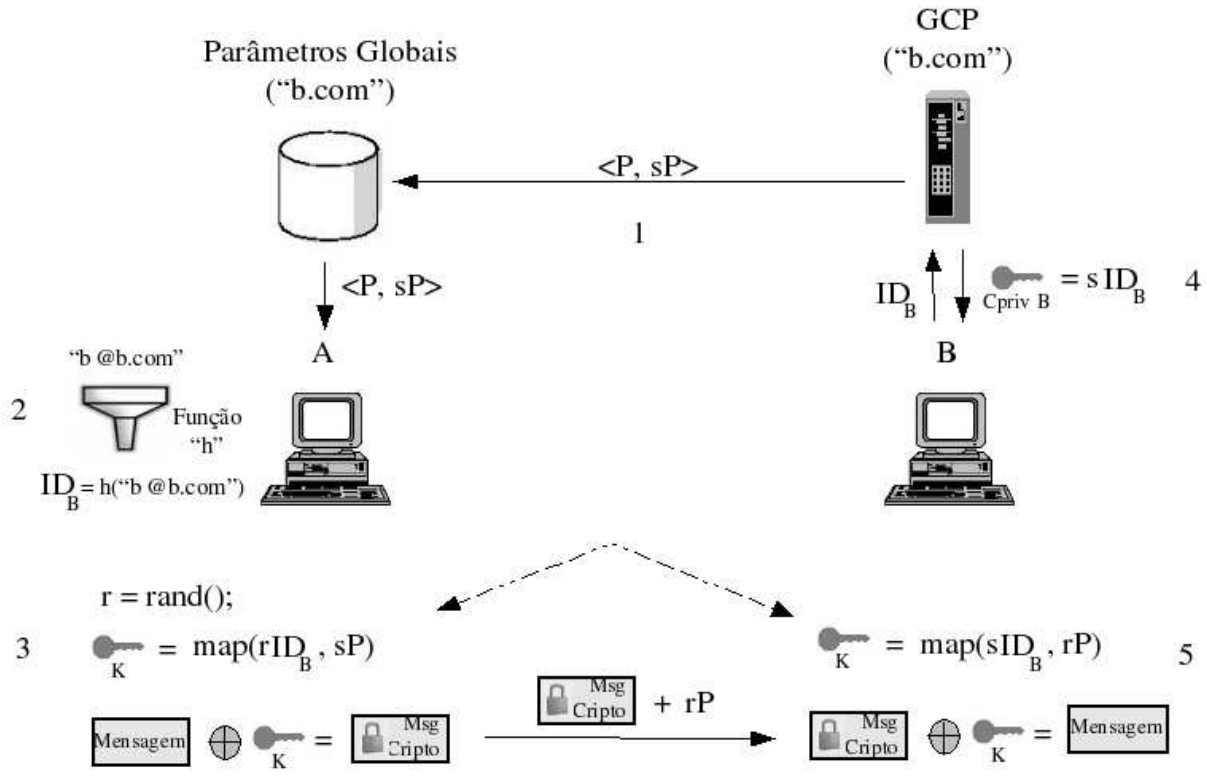

Figura 4.5: Criptografia por Identidades Aplicada à Troca de Mensagem Segura

Passo 1 **Configura**: O servidor de chaves (GCP) escolhe uma chave secreta s (chave mestra) e um ponto P, calcula sP e publica $\langle P, sP \rangle$ para seus usuários;

Passo 2 **Criptografa**: A aplica um *hash* na identidade de B (ex."b@b.com") mapeando-a para um ponto na curva, ID_B;

Passo 3 A gera, aleatoriamente, um valor r, calcula uma chave $k = \langle rID_B, sP \rangle$, que é utilizada na criptografia (simétrica) da mensagem, e envia o texto cifrado junto com rP para B;

Passo 4 B, ao receber o conteúdo cifrado, solicita ao GCP uma chave privada que funcione com sua identificação sID_B;
Extrai: O GCP, então, calcula sID_B e retorna a B como sua chave privada;

Passo 5 **Decriptografa**: B recupera, então, a chave k calculando: $\langle sID_B, rP \rangle = \langle rID_B, sP \rangle$ (restabelece a chave k original).

4.2 Aspectos de Segurança dos Mapas Bilineares

No exemplo apresentado pela Figura 4.5, é observado que a segurança do sistema apresentado está, parcialmente, no não-conhecimento dos pontos rID_B (r gerado aleatoriamente por A) e sID_B (chave privada de B). Essa segurança independe do fato dos pontos que realizam "pares" com esses, no mapeamento em questão, serem públicos (sP e rP, respectivamente). Logo, a chave k, utilizada na criptografia dessa mensagem, somente poderia ser recuperada com o conhecimento de rID_B ou sID_B, e a conseqüente aplicação do mapeamento junto a seus respectivos "pares". Nesse contexto, ainda que um atacante [3] tentasse obter a chave-mestra s (o que daria acesso à sID_B) através dos parâmetros públicos P e sP, o Problema do Logaritmo Discreto em Curvas Elípticas (PLDCE) [4] tornaria essa uma tarefa computacionalmente intratável.

Contudo, essas garantias podem não ser suficientes para outros esquemas envolvendo criptografia por identidades e, mais precisamente, mapas bilineares. Na Figura 4.6, os elementos A, B e C trocam chaves através da aplicação de um emparelhamento $\hat{e}$: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$.

[3] Elemento que planeje a invasão ou violação de um sistema de segurança.
[4] Conforme visto no Capítulo 2.

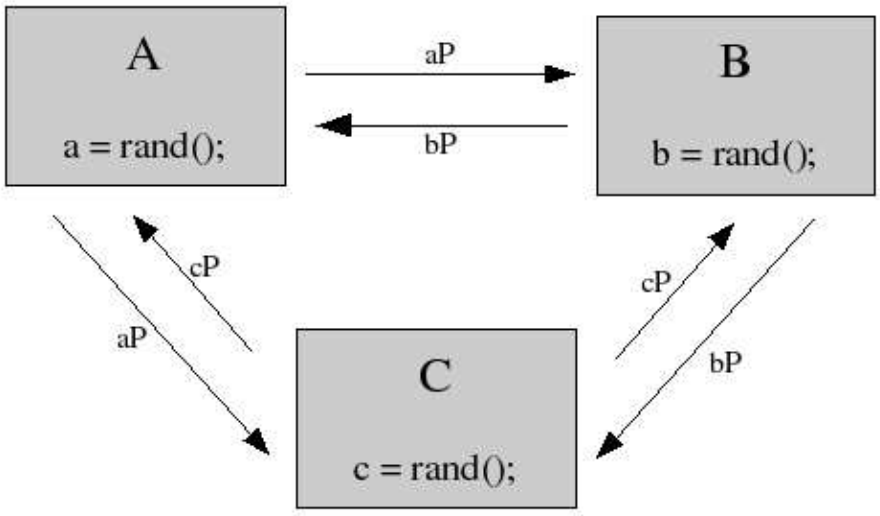

Figura 4.6: Tripla Troca de Chaves com Mapas Bilineares

Passo 1 A gera a e divulga aP;
B gera b e divulga bP;
C gera c e divulga cP;

Passo 2 A calcula $k = \hat{e}(bP, cP)^a = (g^{bc})^a = g^{abc} = \hat{e}(P, P)^{abc}$;
B calcula $k = \hat{e}(aP, cP)^b = (g^{ac})^b = g^{abc} = \hat{e}(P, P)^{abc}$;
C calcula $k = \hat{e}(aP, bP)^c = (g^{ab})^c = g^{abc} = \hat{e}(P, P)^{abc}$;

Passo 3 aP, bP e cP são conhecidos; a, b, c não podem ser obtidos (PLDCE)
$k = \hat{e}(P, P)^{abc} = g^{abc}$ é a chave secreta compartilhada,
P e g são geradores em $\mathbb{G}_1$ e $\mathbb{G}_2$, respectivamente.

Numa análise da Figura 4.6, seria possível imaginar que, mesmo não conseguindo extrair os valores a, b e c (garantia do PLDCE), um invasor poderia, através de um ataque passivo que "escutasse" os valores aP, bP e cP, tentar obter a chave $k = \hat{e}(P, P)^{abc}$. Contudo, isso não é possível graças ao Problema Diffie-Hellman Bilinear (*Bilinear Diffie-Hellman Problem - BDHP*. A seguir, serão definidos esse e alguns outros problemas relacionados, conforme apresentados em [3].

- **DDHP - Decision Diffie-Hellman Problem.** Sejam a, b, c, $n \in \mathbb{Z}_q$ e $P \in \mathbb{G}_1$, onde n é a ordem de P e q é um primo grande:

Problema: Dados P, aP, bP, $cP \in \mathbb{G}_1$, decidir se $c = ab \bmod n$.
Aplicando bilinearidade, tem-se que, para um valor a verificar c, caso $\hat{e}(P, cP) = \hat{e}(aP, bP) \Leftrightarrow c = ab \bmod n$. Assim, o DDHP é considerado um problema "fácil" se aplicado a emparelhamentos "admissíveis", pois tanto o cálculo de $\hat{e}(P, cP)$ quanto de $\hat{e}(aP, bP)$, por definição, devem ser computacionalmente viáveis.

- **CDHP - Computational Diffie-Hellman Problem.** Sejam $a, b \in \mathbb{Z}_q$ e $P \in \mathbb{G}_1$:

  Problema: Dados P, aP, $bP \in \mathbb{G}_1$, calcular abP.
  Como o PLDCE é um problema de "difícil" solução, o CDHP também não apresenta algoritmos em tempo sub-exponencial para resolvê-lo (é computacionalmente intratável).

- **GDHP - Gap Diffie-Hellman Problem.** Seja $\mathbb{G}_1$ um grupo onde o DDHP é tratável, o GDHP consiste em resolver o CDHP para esse grupo $\mathbb{G}_1$.

- **BDHP - Bilinear Diffie-Hellman Problem.** Considerado uma "variante" do CDHP, consiste em:

  Problema: Dados P, aP, bP, $cP \in \mathbb{G}_1$, calcular $\hat{e}(P, P)^{abc}$.
  Assim como o CDHP, trata-se de um problema computacionalmente intratável.

Cabe ressaltar que variações desses problemas acima, denominados "co-problemas" (co-DDHP, co-CDHP, co-GDHP e co-BDHP) aplicam dois grupos "aditivos" distintos ($\mathbb{G}_1$ e $\mathbb{G}_2$) no enunciado de cada um desses "co-problemas". Essas co-variações são úteis ao se analisar emparelhamentos que trabalhem com dois grupos de entrada distintos como, por exemplo, o emparelhamento de *Tate*.

Ao observar a vinculação entre a solução do BDHP e a solução dos demais problemas associados, verifica-se que, a não ser pelo DDHP, trata-se de uma cadeia de problemas computacionalmente inviáveis. Dessa forma, voltando à Figura 4.6,

vê-se que o BDHP serve como garantia de que um atacante jamais conseguiria extrair a chave $k = \hat{e}(P, P)^{abc} = g^{abc}$ somente através da escuta de aP, bP e cP.

É possível concluir, então, que a segurança de muitos dos sistemas baseados em emparelhamentos é uma conjunção dos problemas vistos nesta seção com o PLDCE, proposto pela criptografia por curvas elípticas.

4.3 Vantagens e Desvantagens da IBE

A seguir, serão listadas algumas das principais vantagens e desvantagens associadas à adoção de criptografia por identidades.

Vantagens:

- **Dispensa a existência de um diretório de chaves públicas**. Ao se adotar um padrão de geração de chaves públicas a partir da formação de uma identidade, todos podem aplicar a mesma função na obtenção da chave pública de qualquer elemento do sistema. Isso é uma grande vantagem, devido ao fato de "livrar" o sistema da necessidade de existência de um diretório público e de todos os fatores advindos dessa necessidade, como localização, disponibilidade, gerenciamento e demais aspectos relacionados à manutenção do mesmo.

- **Flexibilidade na escolha de um gerador de chaves privadas**. A criptografia por identidades acaba permitindo que o papel de GCP seja exercido por qualquer elemento de confiança dentro de um grupo. Assim, desde que a esse elemento estejam confiados os parâmetros necessários à geração de todas as chaves privadas, ele pode manter o papel de gerador de chaves dentro de sua empresa ou domínio, dispensando a necessidade de utilização de entidades externas para esse fim. Segundo o exemplo citado em [47], o próprio presidente de uma empresa poderia exercer essa função dentro da sua corporação. Na proposta de extensão SMTP que será abordada no Capítulo 5, esse papel pode ser, opcionalmente, exercido pelo próprio servidor de mensagens.

- **Possibilidade de recuperação de chaves privadas**. O fato do GCP poder gerar, a qualquer tempo, qualquer chave privada dos elementos sob o seu domínio, possibilita que chaves privadas perdidas possam ser recuperadas. Dessa forma, mensagens criptografadas ou mesmo assinadas com base no par de chaves previamente estabelecido podem ser recuperadas ou verificadas, respectivamente. Em alguns casos, o poder de gerar todas as chaves privadas aferido ao GCP pode acabar representando uma garantia de segurança. Em [47], é possível encontrar sugestões de situações críticas, onde uma indisponibilidade parcial ou definitiva de um elemento que não tenha confiado sua chave privada a ninguém podem comprometer a segurança de uma empresa, ou até mesmo a "Segurança Nacional", caso essa entidade venha a se tratar de um órgão de inteligência do Governo. Sistemas assimétricos tradicionais não permitem essa recuperação, pois o principal aspecto de segurança do sistema está justamente na inviabilidade computacional de reestabelecer uma chave privada a partir de sua chave pública correspondente.

- **Não é necessário ter uma chave pública configurada para começar a participação no sistema**. Conforme pode-se observar na Figura 4.5, quando um elemento A precisou enviar uma mensagem para B, não foi necessário que B tivesse requisitado um par de chaves previamente. Dessa forma, o processo de envio de texto cifrado fica bastante simplificado, pois somente quando o destinatário desejar verificar o conteúdo recebido é que o mesmo solicitará ao GCP uma chave privada para essa finalidade. Nesse ponto, dependendo da política de validade de chaves adotada, o destinatário poderá prosseguir utilizando a mesma chave privada na abertura de futuras mensagens a ele destinadas, ao menos enquanto seu par de chaves for válido.

- **Dispensa a obtenção de certificados**. Devido à própria natureza da geração das chaves públicas, onde são utilizados os dados e identificação próprias do elemento participante, deixa de ser necessária a existência de um certificado digital que correlacione uma identidade a uma chave pública associada. Conseqüentemente, fica dispensado o uso da infra-estrutura de chave pública e de toda a complexidade de gerenciamento e demais aspectos relacionados à

manutenção dessas estruturas.

- **Vantagens na aplicação de criptografia com curvas elípticas**. Devido ao fato de aplicar criptografia com curvas elípticas, sistemas baseados em identidades acabam se beneficiando das vantagens relacionadas a essa técnica, de acordo com o que foi visto na Seção 2.4.

Desvantagens:

- **Riscos relacionados à manutenção da chave-mestra no GCP**. A manutenção da chave-mestra do sistema baseado em identidades dentro do gerador de chaves privadas pode trazer riscos relacionados à revelação ou vazamento do seu conteúdo. Uma vez obtida essa chave e, aplicando os demais parâmetros públicos do sistema, qualquer elemento não-autorizado conseguiria produzir as chaves privadas de todos os participantes do sistema. Essa falha de segurança iria perdurar enquanto não fosse descoberto o roubo desse dado, quando, então, o GCP teria que regerar uma nova chave-mestra e divulgar novos parâmetros globais para o sistema. Assim, passa a ser fundamental que a chave-mestra esteja protegida por um sistema bastante robusto a fraudes e invasões. Os trabalhos de [47] e [49] sugerem a aplicação de múltiplos GCPs, de forma a distribuir a responsabilidade da geração de chaves entre vários elementos. Ainda na sugestão de [47] e [49], num cenário com múltiplos GCPs, a segurança do sistema somente estaria comprometida caso todas as chaves-mestras fossem obtidas, pois as chaves pública e privada de cada elemento são um "somatório" de todas as chaves pública e privada administradas por todos esses GCPs.

- **Riscos relacionados a um GCP não-confiável**. Um outro cenário de vulnerabilidade que pode ser aplicado a existência de um gerador de chaves privadas está relacionado a própria idoneidade desse GCP. Um GCP não-idôneo poderia se beneficiar da chave privada de um dos elementos para, por exemplo, forjar assinaturas com a finalidade de produzir falsos documentos.

Nesse aspecto, o trabalho de [50] sugere soluções para situações em que possa ser necessário ter que lidar com GCP "não confiáveis", onde a chave privada de um elemento será composta, além da chave calculada pelo GCP, por um valor aleatório que somente o portador dessa chave poderá saber.

- **Questões associadas a implementação de emparelhamentos**. Fatores como dificuldade de implementação e, principalmente, melhorias de desempenho de emparelhamentos, ainda merecem destaque no universo de desenvolvimento de sistemas baseados em identidades. O trabalho de [46], embora apresente o emparelhamento de *Tate* como uma atraente alternativa, não chega a tornar sistemas baseados em identidades mais rápidos que sistemas assimétricos tradicionais, apesar das vantagens estabelecidas pela utilização de curvas elípticas. Nesse sentido, o trabalho de [45] chega a traçar um comparativo de tempo de execução de esquema de assinatura digital desfavorável à opção baseada em identidades, que chega a ter um desempenho global sete vezes inferior ao RSA. Contudo, trabalhos mais recentes como [51] têm conseguido superar o desempenho do RSA para verificação de assinaturas e tempo de criptografia, através da aplicação de curvas hiperelípticas.

Um balanço geral entre aspectos vantajosos e desvantajosos, mesmo considerando questões de desempenho, cujo processo de pesquisa fatalmente irá reverter, confirma a criptografia por identidades como sendo uma promissora área dentro do escopo de segurança da informação. Segundo [47], sistemas baseados em identidades "podem ser considerados o estado da arte" em criptografia.

Dentro do promissor potencial trazido pelos sistemas baseados em identidades, a próxima seção pretende levantar alguns estudos e variações de aplicação envolvendo essa técnica.

4.4 Estudos, Propostas e Aplicações Existentes

Apesar da proposta de implementação contida neste trabalho concentrar-se na troca sigilosa de mensagens eletrônicas (conforme será visto no Capítulo 5), a lista de possíveis utilizações para sistemas baseados em identidades é bem mais abrangente que isso. As variações de sistemas que utilizam mapas bilineares é ainda mais ampla, pois utilizam a base teórica dos sistemas baseados em identidades aplicada às mais diversas áreas.

A seguir, é possível observar algumas das possíveis variações de utilização dessas técnicas em diferentes cenários.

- **Autenticidade e Marcas D'água**. A dissertação de [52] merece destaque como um interessante exemplo de aplicação de emparelhamento bilinear. Esse trabalho visa analisar algoritmos para localizar alterações em imagens digitais, oferecendo integridade, autenticidade e irretratabilidade, através de "marcas d'água" digitais. Com esse fim, algoritmos que apliquem criptografia assimétrica são utilizadas de forma a gerar assinaturas digitais compactas, as chamadas "marcas d'água". Essas assinaturas são incorporadas aos arquivos de imagens sem, contudo, afetar a qualidade das mesmas. A proposta de [52] apresenta uma aplicação eficiente do emparelhamento de *Tate* no seu trabalho de implementação.

- **Sigilo Programado no Tempo**. Um curioso exemplo de aplicação de criptografia por identidades está no trabalho proposto pelo laboratório da Hewlett-Packard em [53]. Nessa proposta, um servidor de chaves assimétricas gera, periodicamente (a cada minuto), uma chave destinada a abertura de mensagens criptografadas. A criptografia dessas mensagens foi, intencionalmente, construída com uma marca de tempo (data e relógio) específica. Dessa forma, mensagens criptografadas somente podem ser abertas quando o servidor de chaves divulgar, no instante programado, a chave necessária à sua abertura. Observa-se que a aplicação de criptografia por identidades é fundamental na implementação dessa idéia, uma vez que o papel de chave pública é exercido

por uma combinação de calendário com hora de abertura (essa seria a "identidade" no sistema), enquanto a chave privada correspondente é a chave (gerada com base nessa data e horário) periodicamente pelo sistema.

- **Privacidade e Atendimento Médico**. Outro trabalho também proposto pelo laboratório da Hewlett-Packard pode ser visto em [54]. Nesse trabalho, é feita uma breve discussão sobre como a criptografia por identidades pode trazer benefícios dentro de contextos que envolvam serviços de atendimento médico e questões associadas à privacidade e sigilo de informações relativas a pacientes. Vale ressaltar que o trabalho de [54] foi feito com base em uma análise dessa natureza de sistema, que culminou com a sugestão de sistemas baseados em identidades como uma opção barata e com gerenciamento simplificado na solução de questões envolvendo confidencialidade de dados médicos.

- **Cartão de Crédito na Internet**. O trabalho de [55] apresenta uma aplicação do emparelhamento de *Weil* na implementação do uso de cartão de crédito para compras da *Internet*. Esse esquema dispensa a existência de sistemas financeiros durante a confecção de uma negociação eletrônica. Assim, essa proposta garante a autenticidade de um cartão de crédito sem, necessariamente, depender da ajuda e consulta de uma instituição financeira durante uma transação.

- **Esquemas de Assinatura Digital**. Dentro da categoria de propostas para assinatura digital, independente de aplicações direcionadas a um determinado uso, uma variedade de estudos pode ser encontrada. Entre algumas das opções, é possível citar os trabalhos de [56], [57], [50], [58], [59], [60], [61], [62] e [63]. Alguns desses trabalhos apresentam variações de esquemas de assinatura, como assinatura "em anel" e assinatura "cega", cujas descrições podem ser vistas também na dissertação de [47].

- **Propostas de Troca de Chave**. Protocolos de troca de chave baseados em criptografia por identidades são uns dos mais pesquisados temas envolvendo essa técnica. Exemplos como [64] e [65] podem ser citados como referências desse tipo de aplicação envolvendo emparelhamentos de *Weil* e *Tate*, respecti-

vamente. Outros trabalhos também relacionados à troca de chave podem ser vistos em [66], [67], [68] e [69].

- **Protocolos de Autenticação**. Como exemplo de aplicação de criptografia por identidades em protocolos de autenticação, o trabalho de [70] pode ser destacado. Esse trabalho propõe um esquema para eliminação de certificados em transações SSL [5], através da adoção de criptografia por identidades. Em comum com a proposta de implementação contida nesta dissertação, o trabalho de [70] também utiliza a biblioteca IBE [48] em sua implementação. Ainda dentro do assunto envolvendo esquemas de autenticação, o trabalho de [71] também apresenta uma proposta que aplica mapas bilineares nesse tipo de processo.

- **Serviços de E-mail Seguro**. O serviço de *e-mail* seguro é, definitivamente, uma das mais naturais aplicações de criptografia por identidades. Isso se deve ao fato de se poder associar diretamente (ou parcialmente) identidades a endereços eletrônicos. Entre alguns exemplos dessa classe de aplicação, pode ser citada a dissertação de [72]. Já a nível de produto, a solução de *e-mail* seguro oferecida por [4] representa uma opção de mercado baseada em criptografia por identidades.

Uma lista de referências bem mais completa pode ser obtida em [73], onde algumas dessas e cerca de 200 outras citações podem ser encontradas, todas relacionadas aos temas de emparelhamento e sistemas baseados em identidades. Pode-se observar que a recente expansão das pesquisas nessas áreas servem como indicativo de quanto promissor esse assunto pode ser dentro do estudo de segurança da informação.

Finalmente, no próximo capítulo, pretende-se abordar mais especificamente a proposta de implementação contida neste trabalho. Esta proposta será focada no uso de criptografia por identidades dentro do protocolo de transferência de mensagens eletrônicas. Assim, é possível aproximar a aplicação de IBE apresentada aqui a uma variação contida na classe de **Serviços de E-mail Seguro**, citada anteriormente.

[5] SSL - Security Sockets Language.

Capítulo 5

Proposta de Arquitetura IBE com Extensão SMTP

CONFORME comentado na seção 1.1, estudos e propostas que visem a facilitação da geração e troca de *e-mails* seguros trazem novas possibilidades de popularização desse tipo de serviço. De fato, operações de troca segura de mensagens ainda estão muito dependentes dos tradicionais cenários implementados por ferramentas como o *PGP* [1] [74] e o *S/MIME* [2] [75]. Por serem baseadas em geração aleatória de chaves e certificação digital, soluções como essas, embora apresentem estruturas e hierarquias de certificação distintas, vivenciam as mesmas dificuldades advindas da adoção de infraestrutura de chaves públicas, de acordo com o que foi visto na Seção 4.1.1.

A proposta de trabalho que será apresentada neste capítulo, é uma variação de [72], propondo uma arquitetura e implementação de *e-mail* seguro com a conveniência do uso de IBE aliada à transparência resultante da aplicação dessa técnica dentro do protocolo de envio dessas mensagens, o SMTP. Assim, toda a troca de parâmetros, codificação, envio, recebimento e, eventualmente, decodificação de mensagens estarão embutidos no próprio SMTP, de forma transparente aos usuários (tanto remetente quanto destinatário). Somando-se isso à simplicidade com que as chaves

[1] PGP - Pretty Good Privacy.

[2] S/MIME - Secure/Multipurpose Internet Mail Extensions.

públicas dos participantes podem ser geradas (a partir dos seus endereços eletrônicos), espera-se produzir um cenário de fácil aplicação prática, conforme poderá ser comprovado a seguir.

As próximas figuras apresentam uma comparação entre a arquitetura IBE com caminhos para troca de chave e criptografia independentes (Figura 5.1) [3] e incorporados ao servidor de mensagem - contribuição deste trabalho (Figuras 5.2 e 5.3). Nesse sentido, pode-se observar que o processo de criptografia "tradicional", mostrado na Figura 5.1, se dá no nível da aplicação, que se encarrega de copiar (ou pré-instalar) os parâmetros globais do sistema e cifrar a mensagem de texto original. Os processos envolvidos utilizam esses parâmetros e uma chave pública gerada a partir da identificação do destinatário (ex. *b@comp.com*). Os parâmetros globais, nesse caso, precisam ser obtidos a partir do servidor GCP do destinatário, que pode operar acumulando um ou mais domínios de usuários IBE (representados por seus endereços de *e-mail*). Ao receber uma mensagem sigilosa, o destinatário tem que se autenticar com seu servidor GCP para poder obter sua chave privada. De outra forma, não seria possível extrair o texto da mensagem cifrada recebida no mesmo.

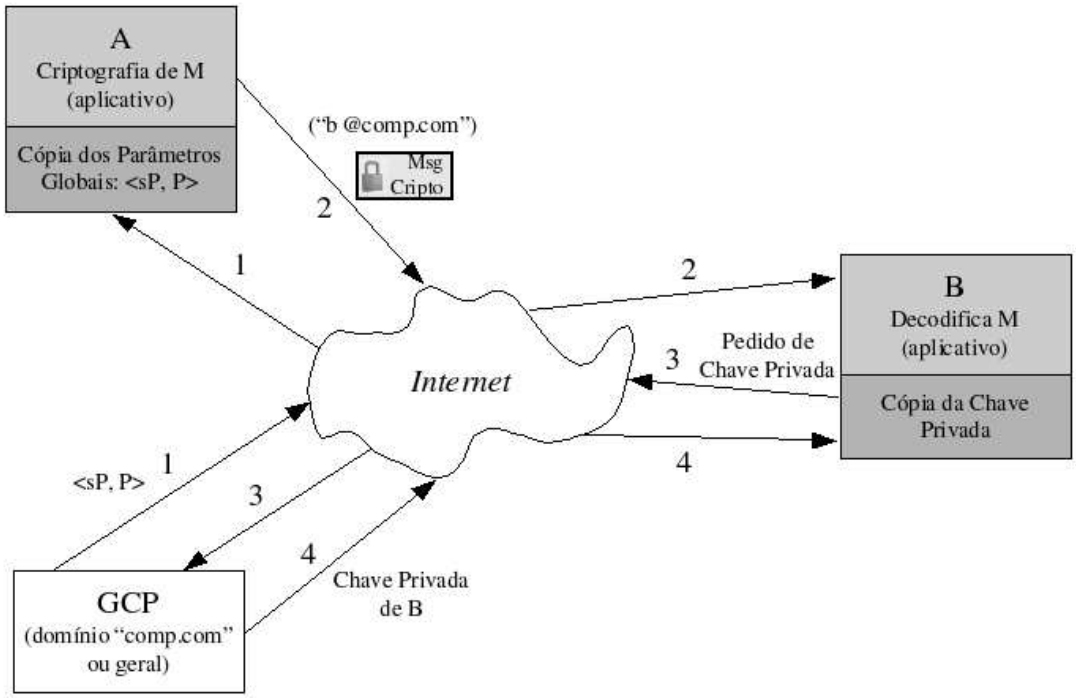

Figura 5.1: Cenário "tradicional" de *e-mail* seguro e IBE

Passo 1 A obtém os parâmetros globais ($\langle sP, P \rangle$) do GCP vinculado a B;

Passo 2 A criptografa a mensagem com base na identidade "b@comp.com" e envia essa mensagem para B;

Passo 3 B solicita ao seu GCP a sua chave privada (caso ainda não a possua);

Passo 4 GCP envia (através de algum canal seguro) a chave privada de B que, agora, pode abrir a mensagem.

Em contrapartida, no cenário proposto na Figura 5.2, tem-se que o próprio protocolo de envio de *e-mail* (no caso, o SMTP) irá se encarregar de negociar os parâmetros globais, específicos do servidor para o qual se deseja enviar a mensagem. Uma vez feita essa negociação, a camada SMTP se encarregaria também de produzir o texto cifrado de forma transparente ao aplicativo. Além dessa transparência, obteria-se também uma maior flexibilidade, presente na adaptação da infra-estrutura de entrega de mensagem segura dentro da própria camada de conexão SMTP. Esse primeiro cenário dessa proposta possibilitaria uma forma totalmente padronizada de operação, onde cada servidor SMTP destinatário teria a responsabilidade pela segurança e manutenção da chave-mestra (s) que viabiliza a geração de chaves privadas para todos os usuários da sua "comunidade".

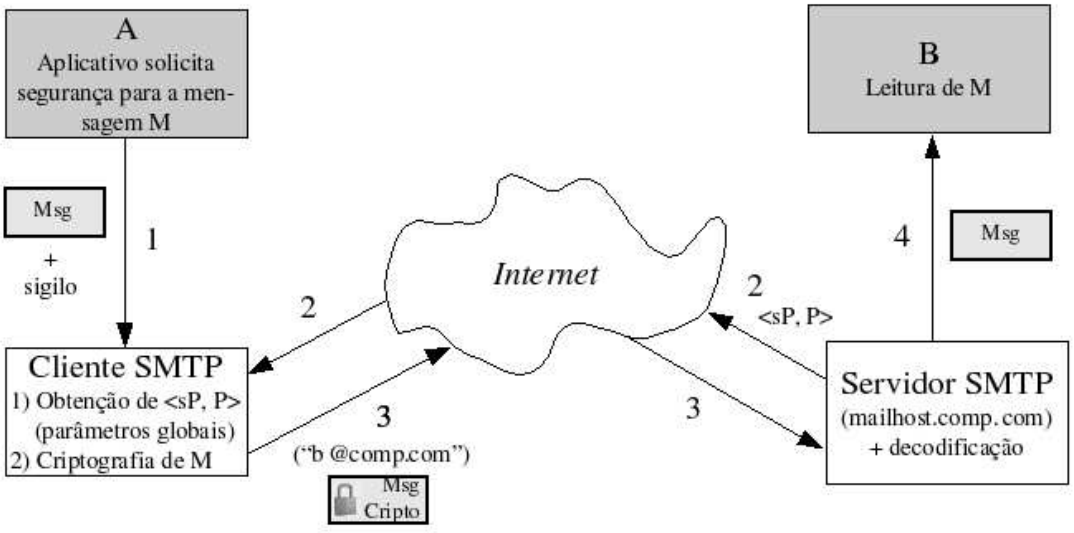

Figura 5.2: Extensão SMTP com IBE e GCP Embutido

Passo 1 A solicita à camada SMTP que a mensagem para B seja sigilosa;

Passo 2 O cliente SMTP de A obtém os parâmetros globais através do servidor SMTP de B, utilizando o próprio protocolo nessa obtenção;

Passo 3 O cliente SMTP criptografa a mensagem e envia a mesma ao servidor;

Passo 4 O servidor SMTP decodifica a mensagem e disponibiliza a mesma a B.

Ainda na Figura 5.2, é possível observar que a proposta deixa em aberto os canais entre o cliente SMTP em ambos os lados (envio e entrega). Assumindo essa flexibilidade, garante-se que, durante o processo de travessia dos dados no âmbito da *Internet*, os dados estariam protegidos pelo processo criptográfico aplicado à mensagem M. O protocolo de entrega da mensagem à B pode, por exemplo, ser uma versão *POP3*[3] [76] com processos de autenticação e criptografia robustos. Nesse contexto, ambos os clientes SMTP (A e B) poderiam, também, estar diretamente ligados à seus servidores, por exemplo, através de um serviço de *WebMail*.

Prosseguindo nessa nova proposta, a Figura 5.3 apresenta uma variação do processo de entrega final e decodificação da mensagem cifrada em B. Na situação mostrada nessa figura, o processo final de decriptografia fica concentrado dentro do próprio destinatário, sendo esse um cenário intermediário entre o visto nas Figuras 5.1 e 5.2. Observa-se também que, nesse cenário "intermediário", o servidor SMTP não mais teria a responsabilidade pela segurança e manutenção da chave-mestra s e, diferente do visto na Figura 5.2, o passo de obtenção da chave privada de B volta a existir.

Finalmente, ainda com relação à Figura 5.3, observa-se que o GCP pode tanto estar vinculado ao servidor SMTP ou ao provedor de serviços desse usuário à *Internet* (provedor de *e-mail*, conexão, etc), quanto a qualquer outro GCP independente, conforme visto na Figura 5.1.

Dessa forma, observa-se nesse novo modelo (Figuras 5.2 e 5.3) a forte ligação entre o servidor SMTP e um único GCP, cujos parâmetros são repassados dentro do protocolo de troca de mensagens. Esse fato sugere a configuração de sistemas IBE

[3]POP3 - Post Office Protocol - Version 3.

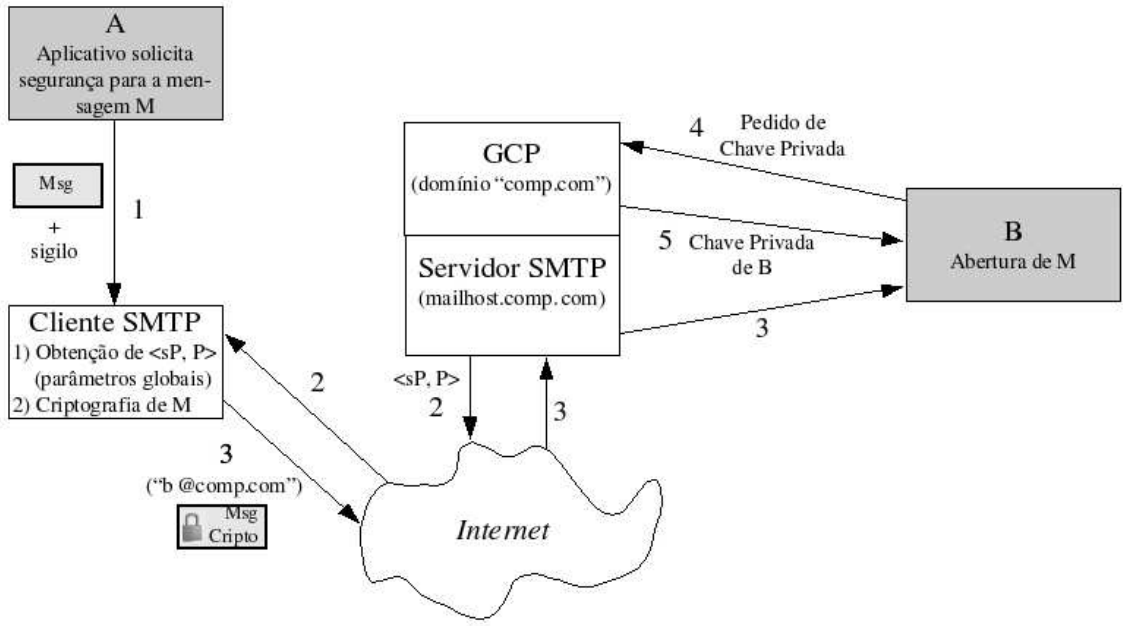

Figura 5.3: Extensão SMTP com IBE e GCP Independente

Passo 1 A solicita à camada SMTP que a mensagem para B seja sigilosa;

Passo 2 O cliente SMTP de A obtém os parâmetros globais através do servidor SMTP de B, utilizando o próprio protocolo nessa obtenção;

Passo 3 O cliente SMTP criptografa a mensagem e envia a mesma ao servidor, que repassa a mesma a B;

Passo 4 B solicita ao seu GCP a sua chave privada (caso ainda não a possua);

Passo 5 GCP envia (através de algum canal seguro) a chave privada de B que, agora, pode abrir a mensagem.

independentes e distintos, e projetados para cada tipo de domínio, como empresas, entidades governamentais, etc. Contudo, diferentes servidores SMTP podem responder os mesmos parâmetros quando, por exemplo, desejar-se configurar um único GCP controlando mais de um domínio (por exemplo, um único GCP controlando diferentes unidades governamentais, como ministérios e repartições públicas relacionadas).

As próximas seções irão tratar de forma mais detalhada o projeto e implementação dessa extensão SMTP, bem como recursos, ferramentas, composição e resultado

obtido até o presente momento. O código fonte produzido (e alterado), bem como arquivos auxiliares na geração do sistema estão disponíveis, respectivamente, no laboratório RAVEL (Redes de Alta Velocidade) da COPPE/UFRJ.

5.1 Decisões de Projeto da Proposta

Como, nesta seção, estarão sendo descritas decisões de projeto envolvendo uma nova versão de protocolo SMTP com IBE integrado, vale a pena apresentar uma breve revisão sobre algumas definições básicas e objetos envolvidos no cenário de troca de mensagens eletrônicas.

5.1.1 Correio Eletrônico e SMTP

O serviço de correio eletrônico foi criado em sua primeira versão em 1972 por Ray Tomlinson. Desde sua criação, muitas arquiteturas de sistemas e protocolos de comunicação associados têm sido propostos, sendo que alguns tornaram-se padrões, estando presentes em inúmeros cenários e aplicações reais.

De forma genérica, existem, dentro do escopo de correio eletrônico, os seguintes mecanismos básicos:

Mecanismo de Transporte é o responsável pelo transporte da mensagem desde o *spool*[4] de saída do remetente até a fila de mensagens do destinatário, dentro do servidor de *e-mail* onde esse destinatário tem conta. Esse mecanismo, freqüentemente, atravessa redes de largo alcance, fazendo o papel de conexão ponto-a-ponto entre o cliente e o servidor de transporte. Como exemplo desse protocolo tem-se o próprio SMTP [7][8].

Mecanismo de Entrega é o responsável pela entrega final de mensagens que chegam dentro do servidor de *e-mail* onde esse destinatário tem conta. Esse mecanismo, basicamente, "move" a mensagem de dentro da fila de mensagens

[4] Arquivo ou área temporária de mensagens.

do usuário no seu servidor para um depósito ou área local (nesse servidor ou em qualquer outra máquina de acesso local) implementando, normalmente, algum mecanismo de autenticação. Como exemplos de protocolos de "entrega final", tem-se o POP3 [76] e o IMAP [77].

De acordo com as definições acima, é possível observar que o papel realizado pelos mecanismos de transporte é fundamental dentro da infra-estrutura de troca de mensagens eletrônicas, compondo a estrutura mínima necessária ao funcionamento desse tipo de serviço. Contudo, desde a criação do correio eletrônico, diversas propostas de padrão de protocolo de transporte de mensagens têm sido propostas. Nesse âmbito, entre as mais populares propostas de protocolo de correio eletrônico estão o X.400, especificado em conjunto por [78] e [79], e o próprio SMTP.

Observa-se que, hoje em dia, o protocolo SMTP tem se mostrado um padrão "de fato", com um maior nível de popularidade, ficando o X.400 mais restrito a algumas áreas geográficas, como Europa e Canadá. Em [80], pode-se observar de forma mais detalhada estudos comparativos entre esses protocolos. De uma forma ou de outra, qualquer que seja o protocolo adotado, conforme pode ser visto em [81] e na próxima seção, ainda existirão formas de conectar esses diferentes ambientes, utilizando os chamados *gateways* de transporte.

O SMTP começou a tomar forma de protocolo padrão para transporte de mensagens em 1982, quando foi iniciado o trabalho de formalização do mesmo, através de [7]. Desde então, diversas revisões foram propostas (entre elas [8]) com o intuito de melhorá-lo e adequá-lo às novas e crescentes necessidades.

Por se tratar de um projeto tão antigo, criado ainda com o a cultura de "login em servidor", o SMTP ainda guarda resquícios e limitações, muitas delas de cunho histórico. Entre essas limitações e restrições, algumas ainda estão remanescentes nas atuais versões e implementações do protocolo, como, por exemplo, a obrigatoriedade da utilização de caracteres ASCII (imprimíveis) no corpo de suas mensagens.

A seguir, serão listadas algumas definições de objetos presentes no padrão e cenário de troca de mensagens via SMTP. Essas definições irão ajudar a entender,

junto com as figuras apresentadas mais adiante nesta seção, as bases desse protocolo.

Agente de Usuário (também denominado *MUA - Mail User Agent* [8]) é o elemento responsável por auxiliar o usuário a enviar e receber mensagens eletrônicas dos servidores SMTP. Como exemplos de agente usuários, tem-se: *mutt*, *elm*, *Pine*, *Netscape Communicator* e *Outlook*, somente para citar alguns. *MUAs* geralmente são utilizados para conectar uma máquina local a algum servidor SMTP, seja para envio ou recebimento de mensagens, além de fornecer uma interface amigável entre o usuário e a infra-estrutura de transporte à qual este usuário está conectado. Agentes usuários podem utilizar mecanismos locais (ex. *pipes*, no caso de ambiente UNIX) para enviar ou receber mensagens de servidores. De outra forma, o próprio SMTP (no caso de implementações em ambiente *Windows*) é necessário para enviar mensagens e, no caso de recebimento, protocolos de entrega como POP3 [76] ou IMAP [77] também podem ser utilizados. A Figura 5.4 apresenta essas variações de agentes usuários.

Agente de Transferência (também denominado *MTA - Mail Transfer Agent* [8]) é o elemento responsável por consumir as mensagens disponibilizadas para envio pelos agentes usuários, encaminhando-as ao agente de transferência (servidor SMTP) vinculado ao usuário destinatário da mensagem. Também é possível observar através da Figura 5.4, o papel desses agentes no cenário de transferência de mensagens eletrônicas. Segundo [8], existem variações de agentes de transferência que podem ser necessárias quando o *MTA* alcançado não é o agente de transporte associado ao destinatário final da mensagem. Serviços de *relay* ou *gateways* podem ser utilizados no caminho entre o *MTA* origem e destino, conforme pode-se observar na Figura 5.5.

Ainda com relação à Figura 5.5, é possível observar que *gateways* SMTP realizam conversões e desconversões de mensagens entre diferentes sistemas de transporte. Conforme definido em [8], esses *gateways* também podem realizar o papel de *firewall* (alterando o endereço de destino da mensagem), mesmo que somente

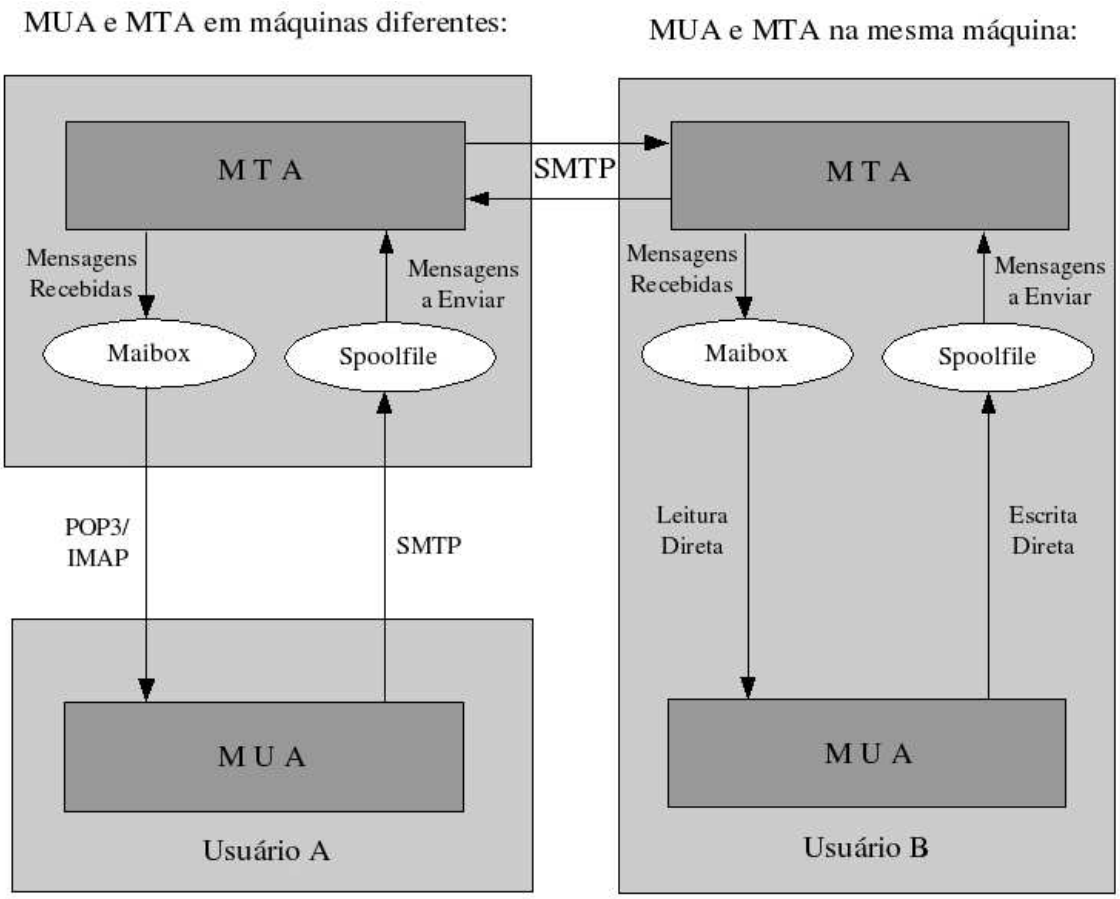

Figura 5.4: Agentes usuários e de transporte

usem SMTP em sua entrada e saída. Por outro lado, serviços de *relay* são realizados exclusivamente por elementos que se comuniquem somente via SMTP e que, por configuração prévia, aceitem o papel de repasse de mensagens a um sistema e entrega. *Relays* SMTP normalmente inserem informação de *trace* na mensagem original sem, contudo, modificá-la.

5.1.2 Extensões SMTP

Com o intuito de viabilizar revisões e melhorias nesse protocolo, o SMTP já prevê a criação de variações, inclusive possibilitando a definição de novos comandos e parâmetros. Essas variações são denominadas *SMTP Extensions* e estão previstas

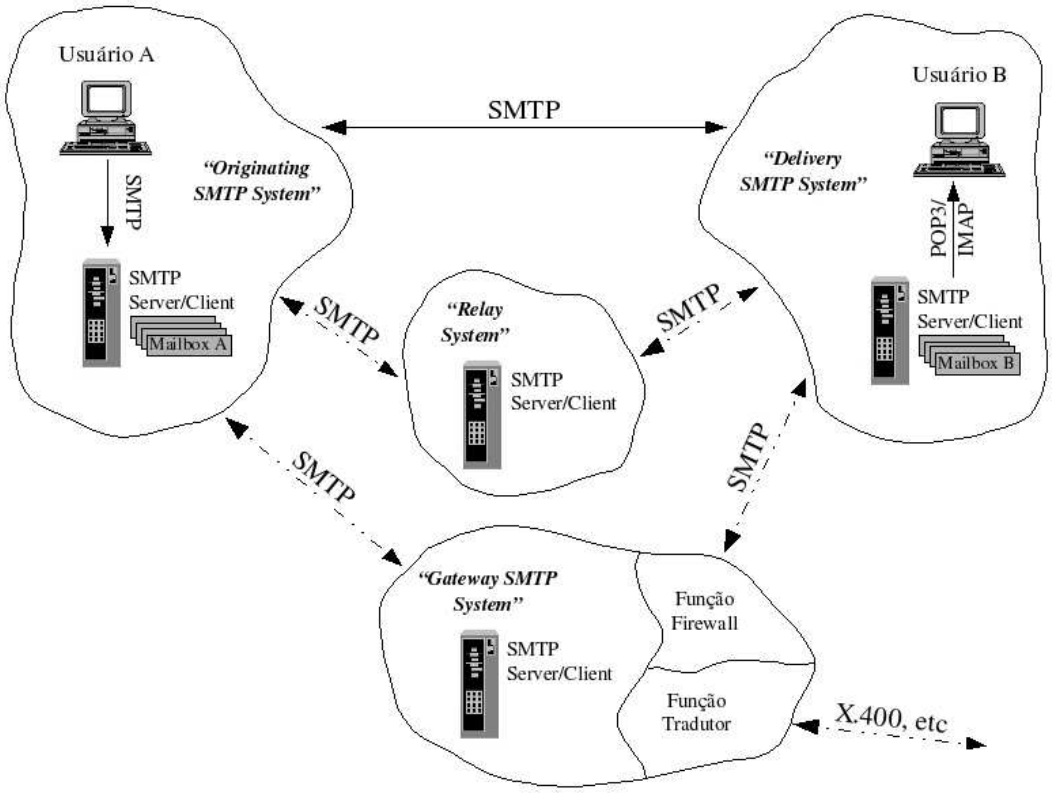

Figura 5.5: Sistemas SMTP conforme definidos em [8]

na RFC [5] [8]. Extensões SMTP possuem, inclusive, padrões recomendados para sua proposição, conforme pode-se constatar em [9] e [82].

Como exemplos de variações dessa natureza, podem ser citadas versões de extensão SMTP com privacidade e autenticação, como em [83] e [84]. De acordo com o que será visto na Seção 5.2, a proposta de extensão SMTP sugerida aqui também buscou seguir o recomendado por [9] e [82].

5.1.3 Proposta: Extensão SMTP com IBE

A idéia de desenvolver uma arquitetura de *e-mail* seguro na camada de transporte dessas mensagens tem como principal objetivo uma simplificação maior do processo de troca de parâmetros, tradicionalmente dependente de entidades externas "confiáveis". Assim, esse modelo assume que o próprio servidor SMTP pode

[5] Request For Comments.

ser considerado "confiável" suficiente para esse fim. Conforme visto no Capítulo 4, na criptografia por identidades, mesmo que um destinatário não necessite ter uma senha pública pré-cadastrada em algum servidor ou diretório confiável para receber mensagens, o remetente precisa obter os parâmetros do sistema ao qual o destinatário está "vinculado". Somente após essa obtenção, a codificação da mensagem, ainda no remetente, pode ser efetivamente feita. A idéia de transferir essa troca de parâmetros para o protocolo SMTP implicaria em uma ainda maior "transparência" do processo.

Conforme foi comentado ao definir as variações de agentes (e de acordo com o apresentado nas Figuras 5.4 e 5.5), a comunicação entre um agente de usuário remetente de uma mensagem e um agente de transporte destinatário poderá acontecer direta ou indiretamente, dependendo do *MUA* estar ou não na mesma máquina do *MTA* desse usuário. A Figura 5.6 mostra como a etapa de troca de parâmetros IBE pode ser feita, na forma prevista dentro dessa nova extensão SMTP, diretamente entre o usuário e o servidor *MTA* destino (ou algum servidor de *relay* ou *gateway* pré-configurado). Na circunstância apresentada pela Figura 5.6, a mensagem seria repassada já de forma criptografada, entre possíveis estágios de transferência (eventuais servidores de *relay* ou *gateway*).

Conforme visto no início deste capítulo, o servidor SMTP de entrega (*SMTP Delivery System* [8]) pode, a partir da decisão em aberto dessa arquitetura, fornecer dados já em formato plano[6] - ficando o mesmo responsável pela geração da chave privada do destinatário a partir do endereço do mesmo - ou repassar a mensagem, ainda no formato cifrado, ao destinatário. Na segunda opção, o usuário que recebe a mensagem cifrada passa a depender da requisição de sua chave privada ao GCP (o mesmo que informou o servidor SMTP dos dados repassados por esse protocolo estendido). A previsão dessa possibilidade está sob a forma de um recurso "#define" dentro da implementação entregue com este trabalho. Assim, foi possível gerar dois servidores SMTP com comportamentos diferenciados, a partir do mesmo código fonte, utilizando a mesma extensão SMTP em ambos os casos. Isto será visto com

[6] Texto em formato aberto ou legível (*plain text*).

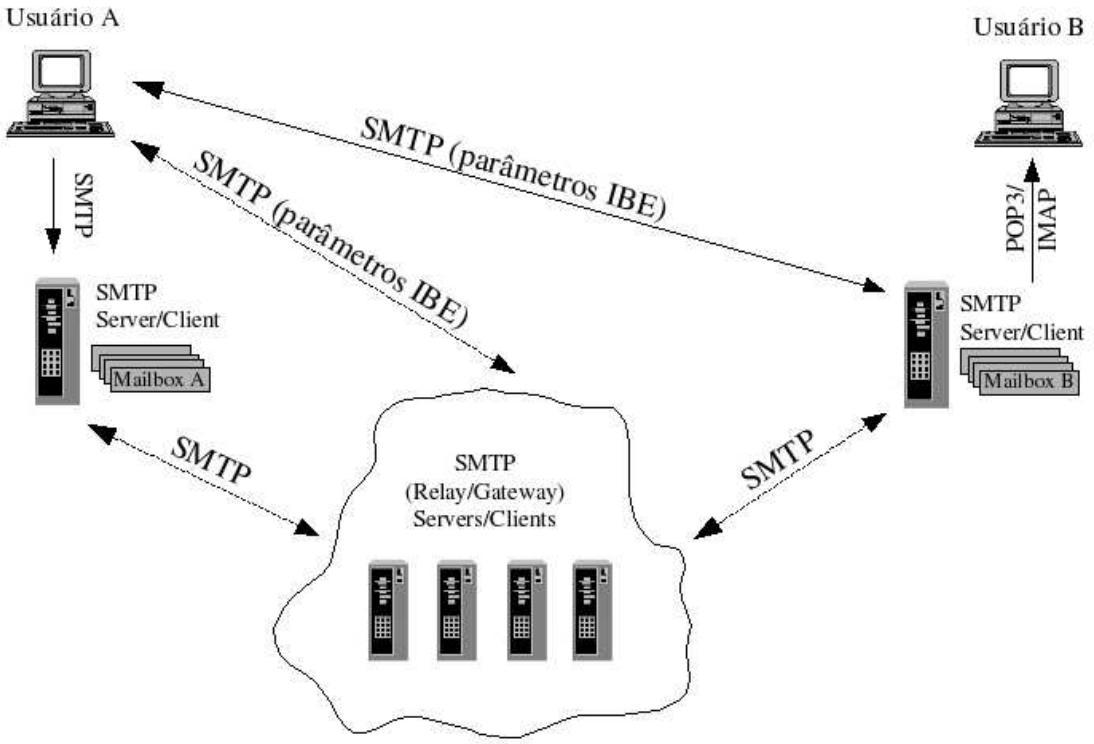

Figura 5.6: Parâmetros IBE e demais contextos SMTP

maiores detalhes na Seção 5.3 e nas seções subseqüentes.

5.2 Descrição da Extensão

O principal objetivo dessa extensão é oferecer uma forma alternativa de empacotamento e envio de mensagens sigilosas dentro do protocolo SMTP, aliando geração, transporte e, em alguns casos, abertura de mensagens criptografadas trocadas entre os elementos envolvidos. Todo o processo de geração de mensagens criptografadas é baseado em criptografia por identidades IBE, onde a chave pública do destinatário da mensagem é utilizada na criptografia de uma chave simétrica. Essa chave simétrica de seção é, por sua vez, aplicada aos dados sigilosos que se deseja transportar entre as entidades envolvidas e somente poderá ser recuperada dentro de um dos agentes destinatários da mensagem, também através de criptografia por identidades.

Dessa forma, agentes/clientes SMTP podem "encapsular" o conteúdo de uma

mensagem sigilosa a partir de:

- Identificação se o servidor SMTP possui ou não o serviço de SMTP estendido com IBE;
- Obtenção dos parâmetros IBE do servidor SMTP do sistema destinatário;
- Opcionalmente, verificação da validade desses parâmetros (discutido na Seção 5.2.6);
- Geração da chave pública do destinatário a partir do endereço de *e-mail* do mesmo;
- Geração de parâmetros locais (chave simétrica cifrada com chave pública e parâmetros globais obtidos) necessários a recuperação dos dados após a entrega final;
- Geração e envio do conteúdo sigiloso junto com parâmetros locais criptografados.

Na outra extremidade, agentes/servidores SMTP ficam encarregados de:

- Fornecimento das variações de serviços SMTP oferecidos, em resposta ao comando de saudação;
- Fornecimento dos parâmetros IBE locais, quando esses forem solicitados;
- Identificação de endereços destinatários parametrizados com dados de criptografia IBE (parâmetros locais gerados pelo remetente);
- Recepção da mensagem e encaminhamento desse conteúdo, junto com os dados necessários à abertura da mensagem, para o cliente final;
- Opcionalmente, abertura da mensagem no próprio servidor de entrega, aplicando a geração automática da chave privada do remetente para recuperação da chave simétrica de seção, quando assim estiver configurado o sistema.

A extensão descrita nesta seção procura seguir o padrão determinado por [9] e [82], conforme pode-se constatar nos itens a seguir.

5.2.1 Terminologia e Convenções da Descrição

Nos exemplos, "C:" e "S:" irão indicar linhas sendo enviadas pelo cliente e pelo servidor, respectivamente.

As palavras-chave que definem níveis de requerimento para RFCs, conforme descritas em [85], serão substituídas nessa descrição por termos como "TEM QUE", "NÃO TEM QUE", "DEVE", "NÃO DEVE", "PODE", "NÃO PODE", "OPCIONALMENTE", "OBRIGATORIAMENTE", etc, de forma a adaptar tais padrões de nomenclatura ao idioma em que esse documento está escrito.

5.2.2 Extensão SMTP com Criptografia por Identidades

- O nome desse serviço de extensão SMTP é "Extensão SMTP com Criptografia por Identidades";
- O valor da palavra-chave "EHLO" associada com essa extensão é "IBECRP";
- A palavra-chave "IBECRP" da resposta "EHLO" não possui parâmetros opcionais, servindo como mera identificação de que esse serviço é suportado pelo servidor SMTP;
- Um novo comando (conforme [8], *verb*) denominado "IBECRP" é definido nessa extensão;
- Um novo parâmetro (opcional) utilizando a palavra-chave "IBECRP" com um valor obrigatoriamente associado, é adicionado ao comando "RCPTO TO".

5.2.3 Identificação da Extensão (comando "EHLO")

Segundo as próprias determinações de [8], caso o cliente inicie uma conexão SMTP com o comando "EHLO" (ao invés da alternativa padrão "HELO") e, caso extensões SMTP esteja implementadas nesse servidor, o mesmo deve retornar uma lista de todos os serviços disponíveis e tratáveis pelo mesmo.

O exemplo a seguir demonstra como é possível identificar as extensões instaladas em um servidor SMTP, através de uma simples conexão via comando *telnet* [7] do UNIX:

```
telnet mailhost.ravel.ufrj.br 25
Trying 146.164.32.67...
Connected to mailhost.ravel.ufrj.br.
Escape character is '^]'.
S: 220 ravel.ufrj.br ESMTP
C: EHLO
S: 250-ravel.ufrj.br
   250-PIPELINING
   250-8BITMIME
   250-SIZE 104857600
   250 AUTH LOGIN PLAIN
C: QUIT
Connection closed by foreign host.
```

Nessa extensão SMTP com criptografia IBE, a resposta a um comando "EHLO" deve incluir a palavra-chave "IBECRP", sem parâmetros opcionais. Assim, o exemplo a seguir ilustra, também através de uma conexão *telnet* na porta 25, a identificação desse tipo de serviço no servidor implementado neste trabalho:

```
telnet dellc400  25
Trying 100.19.19.19...
Connected to dellc400.
Escape character is '^]'.
S: 220 dellc400 ESMTP
C: EHLO
S: 250-dellc400
   250-PIPELINING
```

[7] telnet - Programa que implementa o protocolo TELNET [86] para conexão TCP via terminal.

```
   250-8BITMIME
   250 IBECRP
C: QUIT
Connection closed by foreign host.
```

5.2.4 O Comando IBECRP

O comando "IBECRP" é utilizado pelo cliente SMTP como forma de obter os parâmetros IBE locais ao servidor SMTP destino. Dessa forma, esse comando não possui nenhum parâmetro associado e somente será utilizado pelo cliente ou agente originário da mensagem, sendo dirigido ao servidor SMTP do sistema destinatário.

Em resposta a esse comando, a transferência desses parâmetros por parte do servidor SMTP se dará através da resposta "334 IBECRP server param", seguida pela forma serializada e codificada em "BASE64" [8] dos parâmetros do mesmo (restrições do SMTP [8]). Contudo, a identificação desse comando por parte do servidor não deve significar que a comunicação irá, necessariamente, prosseguir com a transferência de dados cifrados. Assim, o estado interno do servidor não deve ser alterado como conseqüência da recepção do comando "IBECRP".

Uma importante observação em relação a esse comando já foi discutida de forma mais superficial em seções anteriores (Figura 5.6). Os parâmetros IBE do sistema destinatário devem ser diretamente consultados pelo elemento que gera os dados cifrados. Em termos práticos, um cliente SMTP em um agente usuário deve ter acesso direto aos parâmetros IBE do sistema destino, mesmo que utilize outros servidores de *relay* ou *gateway* no processo de transporte da mensagem. Em contrapartida, clientes e servidores intermediários (*relay* e *gateway*, por exemplo) devem estar implementados de forma a repassar os parâmetros locais utilizados na geração do texto cifrado, parâmetros esses que são transferidos através do comando "RCPTO TO". Sem esses dados, as informações contidas no comando "DATA" não podem ser diferenciadas de informações de texto plano e tampouco poderão ser recuperadas no

[8] Formatação de dados binários em seqüência de caracteres "imprimíveis" "A" a "\", que representam os valores binários 000000 (0_{10}) a 111111 (63_{10}), respectivamente.

sistema destino.

Após obter os parâmetros IBE transferidos pelo servidor SMTP, o cliente deve utilizá-los na geração e criptografia de uma chave simétrica de seção, cujos dados de recuperação devem ser transmitidos dentro do comando "RCPT TO". Essa chave de seção deve ser recuperada e, então, utilizada pelo sistema destinatário para decifrar o texto transportado através do comando "DATA".

De volta à Figura 4.5, os parâmetros globais transferidos em resposta a esse comando correspondem, ainda que de forma superficial, ao par de pontos $\langle P, sP \rangle$ contidos naquela figura. O conteúdo e estrutura presente nesses dados, na prática, vai bem mais além desse par de pontos e, para efeito de implementação dessa proposta, irá seguir a parametrização IBE sugerida pela biblioteca IBE [48].

O exemplo a seguir demonstra como acontece uma conexão somente para troca de parâmetros:

```
S: <Aguardando por conexão na porta TCP 25>
C: <Abre a conexão>
C: EHLO dellc400
S: 220 dellc400 ESMTP
   250-dellc400
   250-PIPELINING
   250-8BITMIME
   250 IBECRP
C: IBECRP
S: 334 IBECRP server param
AAgACQAEAEAAFACUAJQABAAEaWJlLTAuNy4ydGVzdDSQbi85GgMqRRgyIEGs1kpE
O75LKALJ1Oiw0aBOsulkWwzc1Lv1px0WEzdHSZQbIssaJffjnB8lj1BhQR5MI9+A
AAAAAAA/////////////////wACAEcARwACAEAAARqAdIqS5TTmjJr/A3yrzpw7
W+dmSk7h1MEXvUgILDlwPkzNVWCpDrgkV+LzcNMiDbMmBcyFIhk1/qTtzCjehbUA
AAIAQAABLmSAJ3frIJF72k4w6EgmGwSuCVpmw6PzSpnAuVc7YoRaZicE++Q2fGDW
fAIOt+uJ98WzIDPjT4Oe8+mMtxlP6wAAAgBHAEcAAgBAAAEPCUW71+mCJjmfz5of
sx3SmWWMa034Ul+hrAJI+MsrVe0l7bZ7IS5q1ovPV8H4mj5xljArSgAfDcEc63Uu
```

```
ww8dAAACAEAAASZncudpQhbvP4tlRf2Bq/zZ7lGkAmJ1i9oGq89jL41IlKbNnmSH
/wABAJIaz88MFOxgoNWkZN6YZCFABCCiY9UAAAAAAAAAAAA=
C: QUIT
```

5.2.5 Parâmetro IBECRP (comando "RCPT TO")

Conforme previsto em [8], comandos padrão do SMTP, como "RCPT TO" e "MAIL FROM" admitem a criação de parâmetros, assim como valores associados aos mesmos. Aproveitando essa possibilidade, essa extensão SMTP criou o parâmetro "IBECRP" dentro do comando "RCPT TO". Esse parâmetro, quando utilizado, necessita obrigatoriamente de um valor associado ao mesmo, valor esse que contém dados fundamentais à recuperação da chave secreta utilizada na criptografia dos dados transportados pelo comando "DATA". Assim como os parâmetros serializados enviados através do comando "IBECRP" e os dados criptografados contidos no comando "DATA", o valor contido nesse parâmetro também é convertido em "BASE64" para poder viabilizar sua transferência dentro das restrições desse protocolo.

Sintaxe:

```
RCPT TO: <caminho_destino> IBECRP=valor_base64
```

Novamente voltando à Figura 4.5, o valor contido nesse parâmetro é análogo ao dado representado pelo ponto rP gerado pelo remetente (naquela figura, A) e repassado ao destinatário (B) no 3^o passo daquele processo. Cabe ressaltar que esse dado é produzido através da aplicação direta da API [9] da biblioteca IBE [48] e, assim como a parametrização do sistema, o mesmo permanecerá, no escopo dessa proposta, vinculado ao estabelecido por [48].

Na existência de um comando "RCPT TO" parametrizado com "IBECRP", o servidor SMTP deve não somente armazenar o valor informado nesse parâmetro, mas

[9] API - Application Programming Interface.

também assumir que o comando "DATA" irá conter, obrigatoriamente, dados criptografados com esse serviço. Clientes/servidores intermediários que implementem essa extensão devem repassar esses dados aos servidores para os quais a mensagem deva ser propagada. Na eventual impossibilidade de entrega da mensagem no formato dessa extensão, esses elementos intermediários devem informar aos clientes esse fato.

O exemplo a seguir mostra como pode se dar uma transferência com sigilo nessa extensão:

```
S: <Aguardando por conexão na porta TCP 25>
C: <Abre a conexão>
C: EHLO dellc400
S: 220 dellc400 ESMTP
   250-dellc400
   250-PIPELINING
   250-8BITMIME
   250 IBECRP
C: MAIL FROM: <root@dellc400>
S: 250 ok
C: RCPT TO: <julio@dellc400> IBECRP=AAIARwBHAAIAQAABCrl4YPze4Aao9rK
utn7F6CcPXkmvhPJTEd0WUlgHKEoa3Hoe
rFdAhbPdF51IMGoayvvSOhBBXiaSkuYw776JYQAAAgBAAAEOnZJNT9zOxciFKhme
CPCEdFPvkpTfs541FXZDf0ODURCGf3jCbmO6kYemxptVsm4uW9jkcA2HRSDM7IaO
b7WIAA==
S: 250 ok
C: DATA
S: 354 go ahead
C: 0Hs/IEbcGkUICrRhxjG5Vs3D9yGGVrWYCXAq1erPW2bTCB30Te8xStU+FBRDxtAj
KMCo8CFZ8ez+WYRhJpZPqUfBtgsmTh2rv30+ZE5ksdNg7vwgHy9f8L8pxL+0Rz8+
eygoqQHzBsPQpI0+efuV4uwNEZDi7VaPMCDZjuQVbilKUMG8giz0xfLkL3cZeNFF
n8L4gy8KLKB17AWekmaP0Q==
```

```
S: 250 ok
C: QUIT
```

No último exemplo, foi possível identificar a aparência da mensagem, já em formato criptografado (e também convertida para "BASE64" - caracteres ASCII), contida dentro do comando "DATA". A seguir, será verificado o conteúdo da mensagem que deu origem a essa codificação, em duas possibilidades de disponibilização no *MUA* (utilitário *mail* do UNIX) do usuário destinatário: Ainda em estado codificado, sendo necessária a obtenção de uma senha privada (e conseqüente aplicação de utilitários de decodificação da mensagem) ou já em formato decodificado, fornecido pelo próprio agente de transporte.

Versão *MTA* sem GCP:

```
From root@dellc400 Sat Jan 15 18:34:24 2005
Delivered-To: julio@dellc400
Subject: This is a SMTP IBE Crypto Message.

>>>> Contact your PKG Server if you still have no private keyword to
open this message <<<<

IBE_SENDER_SESSION_INF

AAIARwBHAAIAQAABCrl4YPze4Aao9rKutn7F6CcPXkmvhPJTEd0WUlgHKEoa3Hoe
rFdAhbPdF51IMGoayvvSOhBBXiaSkuYw776JYQAAAgBAAAEOnZJNT9zOxciFKhme
CPCEdFPvkpTfs541FXZDf0ODURCGf3jCbmO6kYemxptVsm4uW9jkcA2HRSDM7IaO
b7WIAA==

IBE_SENDER_DATA
```

```
0Hs/IEbcGkUICrRhxjG5Vs3D9yGGVrWYCXAq1erPW2bTCB30Te8xStU+FBRDxtAj
KMCo8CFZ8ez+WYRhJpZPqUfBtgsmTh2rv30+ZE5ksdNg7vwgHy9f8L8pxL+0Rz8+
eygoqQHzBsPQpI0+efuV4uwNEZDi7VaPMCDZjuQVbilKUMG8giz0xfLkL3cZeNFF
n8L4gy8KLKB17AWekmaP0Q==
```

É possível observar no exemplo acima, que o preenchimento dos dados e separação dos mesmos em seções foram uma mera decisão de implementação do servidor que repassou essa informação. Não obstante a isso, seria interessante o estabelecimento de um padrão único para a formatação desses dados criptografados, de forma a permitir a confecção de utilitários para decodificação e leitura desse conteúdo. Conforme será visto mais adiante, este trabalho também implementa uma versão simplificada desse tipo de ferramenta: o **qmail-ibedcr**.

A seguir, o conteúdo dessa mensagem, já em formato decodificado, na máquina do destinatário.

Versão *MTA* com GCP embutido:

```
From root@dellc400 Sat Jan 15 18:34:24 2005
Delivered-To: julio@dellc400
Subject: Assunto da mensagem confidencial

Primeira linha da mensagem confidencial
Segunda linha
Terceira e penúltima linha
Última linha da mensagem secreta
```

5.2.6 Considerações de Segurança

Esta seção pretende abordar alguns aspectos relativos à segurança da solução proposta neste capítulo. Dessa forma, estarão sendo discutidos três aspectos básicos:

- **Abrangência de Sigilo.** Irá descrever até que ponto as formas de entrega final da mensagem poderão garantir a confidencialidade de uma mensagem.
- **Resistência a Ataques Passivos.** Tratam situações em que possa haver "escutas" eletrônicas operando somente de forma passiva no ambiente de troca de mensagens.
- **Resistência a Ataques Ativos.** Tratam situações em que um atacante[10] possa estar interagindo ou operando, tanto passiva quanto ativamente (forjando mensagens para ambos os lados), como terceiro elemento dentro de uma conexão.

Abrangência de Sigilo

Voltando novamente às Figuras 5.4 e 5.5, pode-se observar que o SMTP não se trata, necessariamente, de um mecanismo de ligação "fim-a-fim" entre remetente e destinatário. Da mesma forma, a proposta de um SMTP estendido com criptografia IBE não se propõe, necessariamente, a cobrir todos os aspectos de sigilo entre agentes usuários. Isso deve-se ao fato de que, de acordo com o observado na Seção 5.1, a entrega final pode, inclusive, depender da utilização de protocolos diferenciados, como o *POP3* e o *IMAP*.

Contudo, como essa extensão deixa em aberto a questão do servidor de entrega decodificar ou não o conteúdo sigiloso recebido, fica a cargo do administrador do sistema de *e-mail* implantar ou não essa possibilidade. No caso de simples "repasse" dos parâmetros e dados criptografados, o sigilo da mensagem fica assegurado até que o cliente ou agente *MUA* decodifique, ele mesmo, os dados recebidos. Por outro lado,

[10] O termo atacante será utilizado no decorrer deste capítulo com a conotação de intruso ou invasor do sistema.

caso seja implementada a solução integralmente embutida no servidor destinatário, algum outro meio complementar de sigilo deve ser incorporado no caminho entre esse servidor e o agente *MUA*. Logo, diferentes políticas de segurança podem ser aplicadas nos mais diversos casos.

Resistência a Ataques Passivos

Quando, ainda no Capítulo 1, foram tratadas situações em que ferramentas de escuta eletrônica poderiam ser facilmente operadas de forma a quebrar o sigilo de mensagens eletrônicas, procurou-se enfatizar os riscos envolvidos nessa natureza de ataque. Conforme citado especificamente na Seção 4.2 e representado pela Figura 4.5, as garantias trazidas pela segurança envolvendo a aplicação de mapas bilineares e do próprio PLDCE, na forma que foram aplicadas nessa extensão SMTP, fatalmente asseguram que nenhum atacante que aplique escuta passiva poderia violar a confidencialidade dessas mensagens.

Resistência a Ataques Ativos

Outra natureza de ataque que poderia ser aplicada nesse cenário de SMTP com extensão IBE, envolve ataques que possam, de alguma maneira, permitir ao atacante interagir com ambos os lados, interceptando e gerando mensagens entre o cliente e servidor SMTP. Numa situação como essa, o atacante pode se fazer passar pelo servidor SMTP perante o cliente e vice-versa, de forma a obter vantagens que o leve a violar a segurança do sistema. Esse ataque de natureza ativa, também conhecido como *man-in-the-middle* pode ser visto na forma de exemplos reais em [87].

Imagina-se que esse tipo de abordagem poderia ser utilizada pelo atacante visando forjar a transferência dos parâmetros globais para o cliente (se fazendo passar pelo servidor legítimo) e, assim, prosseguir monitorando a conexão objetivando capturar o conteúdo que somente esse atacante poderá decodificar. Mesmo sendo de mais difícil execução quando comparado a ataques passivos, essa natureza de ataque também pode ser evitada no escopo dessa proposta.

Conseqüentemente, é possível citar algumas alternativas para que se possa resolver esse tipo de situação:

- Extensões SMTP com conexão autenticada, como, por exemplo, o SMTP Seguro sob TLS (*Service Extension for Secure SMTP over Transport Layer Security*) [83], podem ser usadas em conjunto com a proposta de extensão feita nesta dissertação, de forma a impedir que esse tipo de abordagem obtenha êxito. A proteção estaria no fato de que uma conexão segura, ao menos para troca de parâmetros de domínio, poderia ser estabelecida entre os legítimos cliente e servidor SMTP do destinatário, ficando o atacante impossibilitado de conseguir se passar por uma ou outra parte. A Figura 5.7 ilustra essa possibilidade.

- Uma outra alternativa de evitar esse tipo de ataque, seria também aplicar criptografia por identidades e assinatura digital nos parâmetros transferidos por qualquer servidor SMTP. Assim, é possível utilizar configurações fixas e de conhecimento geral (instalados em arquivos locais aos clientes SMTP) que, inclusive, podem estar vinculadas a uma estrutura que empregue GCPs múltiplos [50], independentes e geograficamente distribuídos. Conforme citado na Seção 4.3, esse tipo de esquema possibilita que a chave privada do solicitante (nesse caso, o servidor SMTP) não fique no conhecimento de um único GCP, o que confere maior robustez ao sistema. Nessa alternativa, todo cliente SMTP passaria a verificar a validade dos parâmetros recebidos, com base em configurações pré-instaladas e válidas para qualquer domínio. Finalmente, um eventual atacante ficaria impossibilitado de forjar a assinatura do servidor legítimo, visto que seria impossível assinar parâmetros falsos para aquele domínio. A Figura 5.8 exemplifica esse cenário.

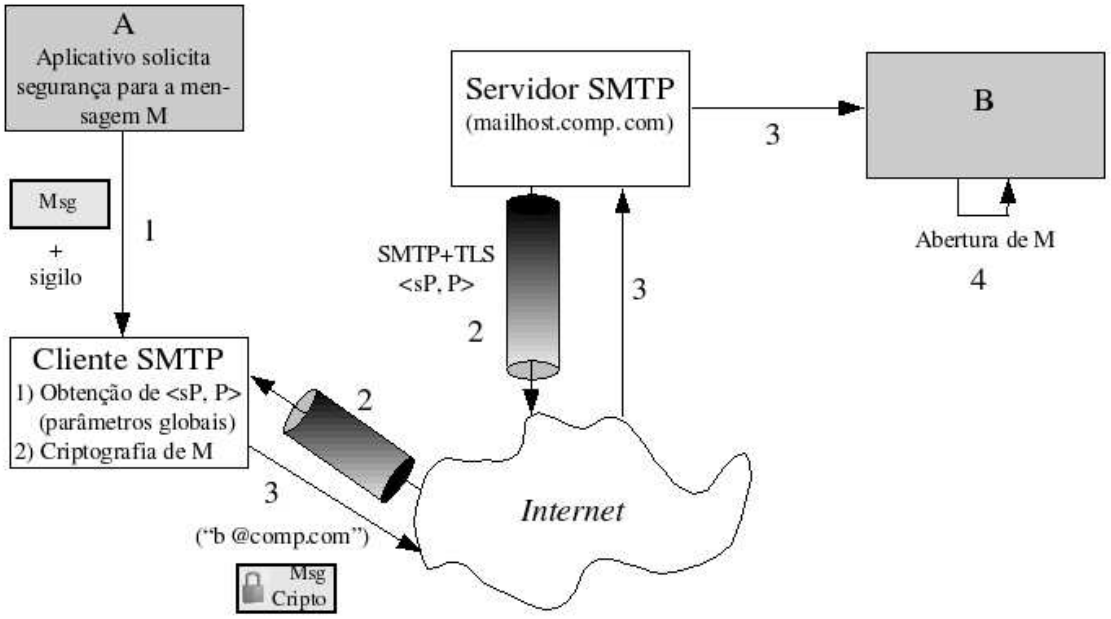

Figura 5.7: Extensão SMTP com IBE e Parametrização TLS

Passo 1 A solicita à camada SMTP que a mensagem para B seja sigilosa;

Passo 2 O cliente SMTP de A obtém os parâmetros IBE do servidor SMTP de B através da aplicação de um canal SMTP seguro como, por exemplo, SMTP+TLS [83];

Passo 3 O cliente SMTP criptografa a mensagem e envia a mesma ao servidor que repassa a mesma a B;

Passo 4 B aplica sua chave privada para abrir a mensagem.

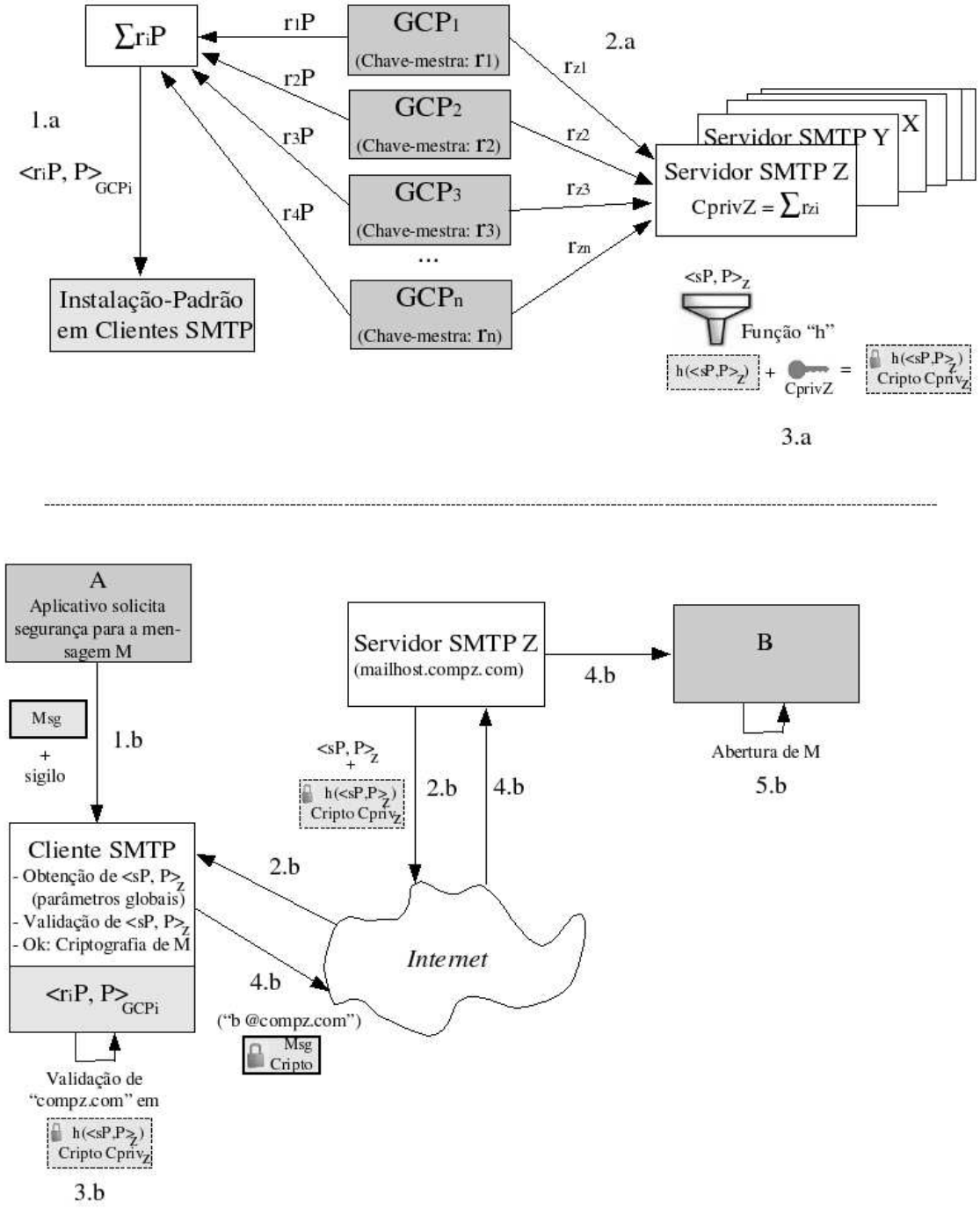

Figura 5.8: Extensão SMTP com IBE e Parametrização Assinada

Passo 1.a Clientes SMTP passam a ter parâmetros locais pré-instalados e previamente gerados pelos GCPs eleitos para esse fim;

Passo 2.a Todos os servidores SMTP interessados em validar seus parâmetros, solicitam partes de suas chaves privadas a cada um desses GCPs, obtendo-as através de canais seguros;

Passo 3.a O servidor do domínio "comp.com" assina seus parâmetros IBE com base na chave privada construída a partir das demais;

Passo 1.b A solicita à camada SMTP que a mensagem para B seja sigilosa;

Passo 2.b O cliente SMTP de A obtém os parâmetros IBE do servidor SMTP de B que passam a conter também a assinatura do mesmo;

Passo 3.b O cliente SMTP valida os parâmetros enviados por esse servidor, através de parâmetros locais válidos para qualquer servidor SMTP;

Passo 4.b Se os parâmetros são válidos, cliente utiliza os mesmos na criptografia da mensagem e envia a mesma ao servidor que repassa a mesma a B.

Passo 5.b B aplica sua chave privada para abrir a mensagem

5.3 Implementação da Extensão

O processo de implementação apresentado nesta seção visa demonstrar uma aplicação real da idéia de variação do protocolo SMTP proposta neste trabalho. Dessa forma, pretende-se descrever os passos e decisões tomadas em todas as fases desse projeto, bem como os componentes de *software* utilizados, como bibliotecas e códigos-fonte abertos. Também serão citados alguns componentes que precisaram ser inteiramente produzidos, dentre os quais estarão módulos utilitários, ferramentas de suporte e de teste.

5.3.1 Decisões de Implementação e Ferramentas Utilizadas

O primeiro passo tomado com relação à implementação contida neste trabalho foi determinar qual ambiente de desenvolvimento utilizar. Nesse ponto, acredita-

se que a adoção de *software* com fonte aberto em ambiente Linux e compilador C/C++ tenha sido um dos principais fatores para se alcançar uma bem-sucedida implementação dessa proposta.

Buscou-se também acelerar o processo de desenvolvimento ao máximo, utilizando, sempre que possível, bibliotecas e recursos prontos ou em desenvolvimento. Assim, bibliotecas matemáticas e criptográficas foram de suma importância durante essa etapa. Da mesma forma, a adoção de um servidor de *e-mail* pronto, também com código-fonte aberto, facilitou de forma significativa a obtenção do resultado desejado. É importante destacar que a utilização do *software* e código-fonte aberto foi feita com total obediência aos direitos de propriedade e de uso e com fins puramente acadêmicos.

Algumas das principais decisões de abordagem e escolhas associadas a implementação dessa extensão SMTP com IBE, bem como algumas justificativas para suas adoções, podem ser vistas a seguir:

Sistema Operacional GNU/Linux utilizado na versão de kernel 2.4.20-8 e distribuição Red Hat 9.0 (versão 3.2.2-5).

Linguagem de Programação C como linguagem-padrão de uso e geração de ferramentas de infra-estrutura no ambiente de desenvolvimento utilizado. O fato do pacote de mensagens eletrônicas original [88] já vir com código aberto nessa linguagem e dessa ser a ferramenta de desenvolvimento "natural" para *software* básico em ambientes UNIX foram determinantes na utilização do C "puro" (ao invés de, por exemplo, utilizar o C++). Logo, buscou-se priorizar o desempenho (menor *overhead*) e a padronização com o código já existente.

Compilador GNU C/C++ , versão 3.2.2, como ferramenta de compilação e geração do código executável produzido.

Biblioteca Aritmética de Múltipla Precisão GNU/MP Lib - GMP [89], na versão 4.1.3, necessária para a geração manipulação de inteiros grandes. Essa biblioteca tem se demonstrado um recurso eficiente na aritmética de valores in-

teiros de grande extensão, fundamentais na implementação de rotinas, recursos e processos associados à criptografia.

Biblioteca de Criptografia OpenSSL [90], na versão 0.9.7d, contendo rotinas de criptografia que reúnem robustez e desempenho, fundamentais para o desenvolvimento de aplicações na área de segurança.

Biblioteca IBE de Stanford [48], na versão 0.7.2, que realiza a implementação de criptografia por identidades na forma em que foi aplicada neste trabalho.

Pacote de Serviços de E-mail Qmail [88], na versão 1.0.3, onde foram alterados tanto o processo servidor SMTP (processo *deamon* **qmail-smtpd** para tratamento de mensagens de chegada) quanto um processo cliente SMTP (processo **qmail-remote**) com a finalidade de implementar essa extensão SMTP. Entre as principais razões para essa implementação basear-se nesse pacote de serviços de *e-mail*, tem-se: Segurança do *qmail* contra invasores de máquina (aliando simplicidade e implementação robusta), confiabilidade quanto a não-perda de mensagens, ótimo desempenho, baixo uso de recursos, entre outras vantagens [88].

5.3.2 Composição da Implementação

Esta seção descreve todos os componentes de *software*, tais como código-fonte, *scripts*, *makefiles* e executáveis, criados e/ou alterados e que integram a proposta de implementação contida neste trabalho. Dessa forma, os componentes listados aqui não completam a lista total de arquivos necessários à geração do pacote *qmail*. Cabe ressaltar que lista total de arquivos que compõem essa implementação é extremamente extensa e não acrescentaria muito caso fosse incluida nesta seção. Contudo, essa listagem total pode ser obtida de [88] (versão 1.0.3).

A Tabela 5.1 apresenta uma estimativa do esforço de geração/alteração de código em relação ao código já existente antes dessa proposta de implementação:

Versão	Arquivos criados/alterados	Linhas (aprox) criadas/alteradas
qmail original (versão 1.0.3)	630	18000
qmail com extensão IBE	16	2750

Tabela 5.1: Software original X produzido ou alterado

A seguir, pode-se ver o nome e a descrição de cada um desses 18 arquivos criados/modificados durante a implementação dessa extensão e presentes na Tabela 5.1.

- **smtp-ibe.h** (arquivo fonte novo, com 68 linhas) - Contém todas as definições comuns aos arquivos utilizados na integração do SMTP com a biblioteca IBE.
- **smtp-ibe.c** (arquivo fonte novo, com 598 linhas) - Contém rotinas utilitárias e adaptações necessárias a implementação da extensão SMTP com IBE no código original do servidor de *e-mails qmail* e nos demais processos de teste criados. Baseia-se nas bibliotecas IBE, GMP e OpenSSL.
- **qmail-remote.c** (arquivo fonte alterado, agora com 637 linhas) - Implementação de cliente SMTP, alterado para testes com a extensão SMTP proposta.
- **qmail-smtpd.c** (arquivo fonte alterado, agora com 813 linhas) - Implementação de processo *deamon* de servidor SMTP, alterado para testes com a extensão SMTP proposta.
- **qmail-ibepkg.c** (arquivo fonte novo, com 148 linhas) - Implementação do gerador de parâmetros globais aleatórios (do servidor de chaves), bem como do gerador de chaves privadas para uma identidade específica. Essa última funcionalidade foi criada com o intuito de possibilitar a utilização da versão do servidor SMTP sem gerador de chaves interno.
- **qmail-ibedcr.c** (arquivo fonte novo, com 365 linhas) - Implementação de ferramenta local para decodificação de *e-mail* criptografado presente no arquivo *Mailbox* de um usuário, utilizando, para isso, os parâmetros globais do sistema

e da chave privada desse usuário. Cabe ressaltar que esse programa também foi criado com o intuito de possibilitar a utilização da versão do servidor SMTP sem gerador de chaves interno.

- **ibe.dat** (arquivo texto novo, com 15 linhas) - Arquivo de configuração dos parâmetros básicos, como caminhos no sistema de arquivos e nomes de arquivos que irão compor a parte configurável da biblioteca IBE.

- **ibeparam.dat** (arquivo texto novo, com 9 linhas) - Arquivo de configuração dos parâmetros globais do sistema IBE, como pontos, escalares e demais parâmetros públicos do sistema IBE. Esse arquivo é gerado automaticamente pela ferramenta **qmail-ibepkg** e o seu conteúdo compõe, literalmente, os dados transferidos entre servidor e cliente SMTP em resposta ao comando definido na Seção 5.2.4.

- **.ibemkey.crp** (arquivo texto novo, com 1 linha) - Contém a chave-mestra utilizada pelo servidor SMTP ou o servidor de chaves (caso o utilitário **qmail-ibepkg** esteja sendo usado) para que, a partir dos parâmetros globais e de uma identidade de destinatário, a chave privada correspondente a essa identidade possa ser extraída. Assim como o **ibeparam.dat**, esse arquivo também é gerado automaticamente pela ferramenta **qmail-ibepkg**.

- **qmail-remote** (arquivo executável) - Versão executável do cliente SMTP, presente na distribuição original do *qmail* [88] e alterado para implementar a extensão proposta. Esse programa é executado através de linha de comando para o envio de mensagens eletrônicas.

- **qmail-smtpd** (arquivo executável) - Versão executável do servidor SMTP, presente na distribuição original do *qmail* [88] e alterado para implementar a extensão proposta. Esse programa é executado sempre que uma conexão é aberta com a porta 25 do servidor de mensagens.

- **qmail-ibepkg** (arquivo executável) - Novo executável que implementa a ferramenta de geração aleatória de parâmetros globais do sistema ou, opcional-

mente, extrai uma chave privada a partir desses parâmetros e de uma identidade escolhida. Esse programa é executado por linha de comando.

- **qmail-ibedcr** (arquivo executável) - Novo executável que implementa a ferramenta de leitura e processamento do arquivo *Mailbox* com as mensagens enviadas a um destinatário. Esse programa também é executado por linha de comando e necessita da chave privada do usuário para poder decriptografar as mensagens contidas nesse arquivo.

- **conf-cc** (arquivo de *script* UNIX alterado, atualmente com 7 linhas) - Arquivo envolvido na compilação do pacote *qmail*, alterado em função da utilização do depurador no ambiente em que esse pacote foi regerado.

- **conf-ld** (arquivo de *script* UNIX alterado, atualmente com 6 linhas) - Arquivo envolvido na linkedição do pacote *qmail*, alterado também em função da utilização do depurador no ambiente em que esse pacote foi regerado.

- **Makefile** (arquivo *makefile* alterado, atualmente com 2251 linhas) - Arquivo necessário à geração do pacote *qmail* na nova versão (novos módulos, bibliotecas, *includes*, etc), assim como à geração dos utilitários confeccionados para teste e suporte à nova extensão.

5.3.3 Relatório Final de Implementação

Como balanço final deste trabalho de implementação, é interessante ressaltar importantes aspectos e observações feitas durante e após seu processo de confecção. Alguns desses aspectos estão relacionados a dificuldades de instalação e configuração de bibliotecas e pacotes utilizados, passando pelo aprendizado de uso desses recursos e, finalmente, pela geração e depuração do *software*, tanto da parte alterada quanto dos programas totalmente produzidos. Contudo, esta seção pretende também destacar alguns resultados obtidos durante testes e execuções da implementação proposta (relatório de medições e avaliações realizadas).

A instalação e configuração dos pacotes GMP, OpenSSL e IBE ocorreu sem

maiores problemas no ambiente utilizado. Com relação a essa fase, somente foram necessárias algumas leves adaptações, devido à diferença de versões de compilador, afetando principalmente o arquivo **Makefile** da biblioteca IBE [48].

Já durante a instalação do pacote *qmail*, dificuldades relacionadas a problemas de compilação do mesmo [88] dentro da distribuição Linux utilizada foram vivenciadas. Outra leve incompatibilidade na instalação padrão do *qmail*, ainda em relação à distribuição listada na Seção 5.3.1, foi o fato do processo de configuração não abordar especificamente como compatibiliza-lo ao servidor *xinetd*, presente na instalação *default* daquela distribuição. Para sanar esse problema, bastou que fossem utilizados parâmetros de configuração alternativos aos sugeridos no arquivo "INSTALL" desse pacote [88].

Com relação ao aprendizado de uso e manuseio dessas ferramentas, foi sentida a necessidade de uma maior documentação da biblioteca IBE. Dessa forma, foi preciso restringir-se unicamente à leitura de código e à busca de arquivos-fonte com exemplos, incluídos no pacote original de instalação. Contatos via *e-mail* foram realizados com o desenvolvedor de [48], quando, então, surgiu a informação de que uma nova versão está em fase de finalização e logo estará disponível.

O código do pacote *qmail* que foi, efetivamente, a base da implementação dessa extensão SMTP, teve que ser minimamente entendido, uma vez que trata-se de um *software* bastante extenso e com diversas características de instalação e customização. A leitura e manutenção do mesmo também não foi uma tarefa simples, principalmente devido à pouca existência de comentários e ao estilo de codificação utilizado no mesmo, muito compromissado (e notoriamente bem-sucedido) com os quesitos desempenho e robustez. Esse fato acabou representando um desafio a mais na alteração e aproveitamento desses códigos, que tiveram muitas de suas rotinas refeitas e alteradas, não somente com relação à inclusão de novas funcionalidades, mas também melhoria, dentro do possível, de comentários, sempre tentando optar por um estilo de codificação mais claro e legível. Apesar desses esforços, boa parte do código alterado ainda necessita de incrementos com relação a sua legibilidade.

É possível ressaltar, ainda com relação a esses arquivos-fonte, que tanto o código

alterado quanto o produzido não possuem, na versão implementada neste trabalho, grandes compromissos quanto a robustez a eventuais ataques que possam ser destinados ao servidor de *e-mail* gerado por esse código. Dessa forma, não existe nenhum comprometimento quanto à utilização desse servidor em ambientes de produção que possam estar sendo visados com abordagens destrutivas de uso, ficando o código como uma mera prova de conceito das idéias apresentadas aqui.

5.3.4 Relatório de Medições e Avaliações Realizadas

Esta seção pretende concluir o capítulo apresentando dados com medições e comparativos resultantes da execução do código produzido nesse trabalho. As medições aqui contidas dão enfoque a uma análise comparativa de desempenho, concentrando-se no tamanho da mensagem e, principalmente, no tempo médio de execução. Conforme será possível constatar, essas medições pretendem não somente confrontar o desempenho dessa implementação em relação à implementação original de [88], mas também avaliar as diversas opções de execução (e configuração) disponíveis nesse novo código.

Cabe ressaltar que os valores médios de tamanho e tempo de execução que serão apresentados nas tabelas e gráficos a seguir, foram obtidos através da utilização de mensagens de tamanho fixo, contudo, com o cuidado de experimentar os mais diferentes conteúdos para as mesmas. Dessa forma, constatou-se que a variação de formato ou estrutura interna dos dados não chega a afetar significativamente as medições efetuadas.

Análise do Impacto no Tamanho da Mensagem

Conforme visto anteriormente neste capítulo, a codificação "BASE64" foi aplicada a todas as informações trocadas dentro da conexão entre cliente e servidor, como parâmetros do servidor - em resposta ao comando "IBECRP" e dados de obtenção de chave, cujo valor acompanha a proposta de variação para o comando "RCPT TO". Diferentemente do conteúdo presente no comando "DATA", essas informações não

representam texto criptografado, mas contêm dados binários e, como tais, devem ser codificados para que o protocolo possa suportá-los. Contudo, ao criptografar-se o corpo de dados de uma mensagem eletrônica - transportados pelo comando "DATA", independente do conteúdo original ser um texto ASCII ou um conteúdo binário (nesse caso, já codificado em ASCII), não há nenhum controle sobre o fato do resultado da criptografia retornar uma seqüência binária.

De fato, a implementação de criptografia DES aplicada neste trabalho produziu, a partir dos textos utilizados nos experimentos realizados, seqüências com dados binários em seu conteúdo. Assim, viu-se também a necessidade de aplicar, tanto na entrada quanto na saída do processo de criptografia assimétrica, a codificação "BASE64", de forma a converter e desconverter dados criptografados para caracteres ASCII.

As medições de tamanho e crescimento de mensagem realizadas mostraram o custo associado ao emprego dessa codificação e evidenciaram o impacto causado pela codificação de um texto criptografado em caracteres ASCII. Assim, a Tabela 5.2 apresenta o tamanho correspondente ao crescimento desses dados provocado, claramente, pela estratégia de codificação utilizada.

Tamanho Original (bytes)	**Tamanho Final Criptografado (bytes)**
10	25
10^2	157
10^3 $(1K)$	1365 $(1,37K)$
10^4 $(10K)$	13553 $(14,6K)$
10^5 $(100K)$	135428 $(135K)$
10^6 $(1Mb)$	1354178 $(1,35Mb)$
5.10^6 $(5Mb)$	6770847 $(6,77Mb)$
10^7 $(10Mb)$	13541678 $(13,5Mb)$

Tabela 5.2: Tamanho final das mensagens criptografadas para transporte

É possível constatar na Tabela 5.2, que a expansão do tamanho dos dados úteis transmitidos a partir de mensagens superiores a 10^3 bytes é em torno de 35%. Esse percentual deve-se ao próprio método de codificação ASCII utilizado, uma vez que o "BASE64" codifica conjuntos de 6 bits de informação em 8 bits de dado (ASCII) transmitido. Assim, o aumento de cerca de 33.33% é devido ao desperdício de 2 bits a cada 6 bits de dados úteis. Os cerca de $1,6\%$ restantes justificam-se pelo fato dessa codificação inserir alguns caracteres indicadores de tamanho dos grupos de bits codificados (exemplo: "=" e "==") e eventuais "sobras" dentro da própria seqüência de caracteres ASCII gerada.

É interessante frisar que os dados apresentados na Tabela 5.2 somente contabilizam o crescimento na área de dados da mensagem (comando "DATA"), não refletindo a sobrecarga total de informação transferida em toda a conexão cliente-servidor. Isso deveu-se ao fato de se estar procurando enfocar o impacto do crescimento da informação "útil" antes e depois de sua transferência, já como dado criptografado.

Análise do Impacto no Tempo de Execução

Antes de começar a apresentar os dados de tempo de execução obtidos, é importante indicar as condições de teste, recursos de *hardware* e *software*, bem como as demais informações relevantes associadas a essas execuções. Dessa forma, a Tabela 5.3 contém um resumo dessas configurações.

O primeiro comparativo de valores médios de tempo aferidos está apresentado na Tabela 5.4. Essa tabela mostra que não há nenhum impacto relevante no tempo de execução entre a versão original e a modificada, enquanto não se aplica a extensão de criptografia por identidades nas conexões.

A Figura 5.9 representa esses valores em forma gráfica e, assim como os demais gráficos, utiliza escala logarítmica no eixo Y (tempos), para um melhor comparativo entre as diversas ordens de grandeza de tempo obtidas. Cabe ressaltar que, embora não exercite nenhuma situação nova perante a versão de [88], essas medições serviram para ratificar de que o novo código mantém, aproximadamente, o mesmo

Hardware	UCP	Pentium III 1.2GHz
	Memória RAM	256Mb
	Disco Rígido	30Gb
Software	Sistema Operacional	GNU/Linux
	Kernel	2.4.20-8
	Distribuição	Red Hat 9.0 3.2.2-5
Estratégia	Uma única máquina foi utilizada na medição, fazendo os papeis de cliente e servidor. Assim, objetivou-se, principalmente, isolar o atraso gerado por transmissões em uma rede.	
Intenção	Medir os tempos de transferência de dados dentro do cliente (criptografia e transferência do conteúdo do comando "DATA") e tempo de conexão total do lado do servidor (ocupação do mesmo). Dessa forma, serão utilizadas as notações "Tx Cliente" e "Cx Servidor" para representar um ou outro caso, respectivamente, nas tabelas e gráficos a seguir.	
Unidades	Tamanho	Bytes
	Tempo	Milisegundos
Escala	Tempo em base logarítmica (base 10)	

Tabela 5.3: Ambiente de execução, aferição e apresentação utilizado

desempenho original, dadas as mesmas condições de conexão (ou seja, sem o uso da extensão SMTP proposta).

Dando continuidade a avaliação do código produzido, foram realizadas medições dos tempos de transferência e conexão para situações em que haja criptografia, ou seja, utilizando a extensão proposta neste trabalho. A Tabela 5.5 apresenta os valores obtidos para esses tempos em duas situações distintas: Aplicando somente criptografia, feita do lado do cliente, ou realizando criptografia e decriptografia, essa última dentro do próprio servidor SMTP.

Tamanho da Mensagem (bytes)	Servidor Original		Servidor Modificado (sem cripografia)	
	Tx Cliente	Cx Servidor	Tx Cliente	Cx Servidor
10	0,86	4,80	0,87	6,47
10^2	0,88	4,84	0,89	7,43
$1K$	0,89	5,02	0,96	8,29
$10K$	1,49	35,37	1,53	38,27
$100K$	7,64	36,35	7,63	38,86
$1Mb$	189,99	300,69	188,75	347,91
$5Mb$	1016,83	1508,28	1013,14	1446,28
$10Mb$	2214,15	2943,66	2171,19	2838,58

Tabela 5.4: Tempos médios para SMTP sem criptografia

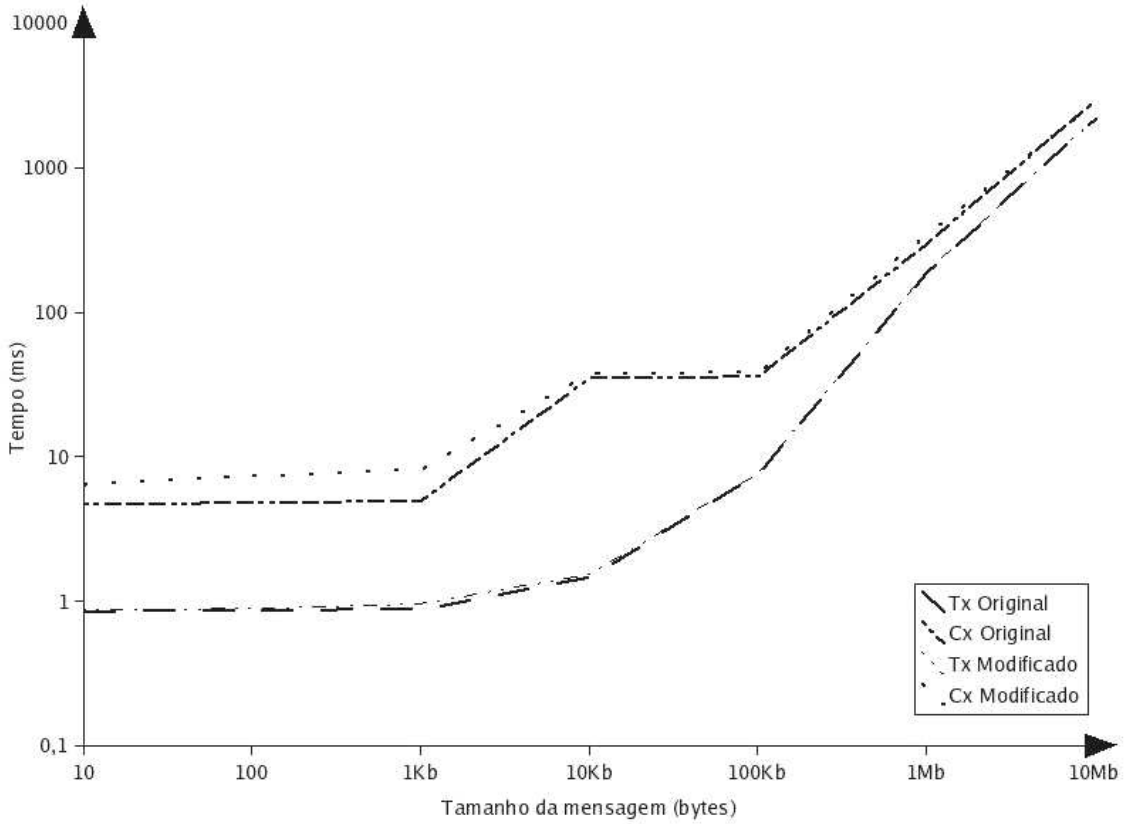

Figura 5.9: Comparativo antes x depois para SMTP sem criptografia

Tamanho da Mensagem (bytes)	Servidor Modificado			
	cripto (somente)		cripto & decripto	
	Tx Cliente	Cx Servidor	Tx Cliente	Cx Servidor
10	1,44	144,46	69,79	213,15
10^2	1,48	144,77	69,90	213,26
$1K$	1,97	176,64	70,60	214,20
$10K$	4,70	177,81	73,55	251,39
$100K$	43,59	208,96	147,43	314,16
$1Mb$	656,21	929,13	769,97	1440,61
$5Mb$	3851,65	4699,79	4398,05	6216,11
$10Mb$	7824,35	8919,15	11027,19	12680,39

Tabela 5.5: Tempos médios para SMTP com criptografia IBE

Para se poder analisar a carga inserida pelas rotinas de criptografia, as Figuras 5.10 e 5.11 apresentam, separadamente, as medições de tempo de transferência e de conexão, respectivamente. Verifica-se em ambas as figuras, que os valores obtidos nessa segunda fase de medição são confrontados com os valores coletados durante os testes sem criptografia, contidos na Tabela 5.4.

Com relação à Figura 5.10, pode-se notar as variações no tempo de transmissão de dados causadas pela incorporação de criptografia no cliente (somente) e dessa criptografia no cliente seguida pela decriptografia no servidor. O aumento no tempo em que o cliente transfere os dados, quando comparado à transmissão "insegura", apresentou um comportamento "linear" a partir do aumento no tamanho da mensagem. Nesses termos, foi contabilizado um aumento médio de 3 vezes do tempo obtido durante as transferências sem criptografia.

Ainda observando a Figura 5.10, ao se comparar transmissões "inseguras" com as contempladas com ambos os processos de criptografia e decriptografia, nota-se que a aparente "linearidade" no aumento de tempo, notada na comparação anterior, deixa de existir. Diferentemente do visto no processo único de criptografia, para mensagens pequenas e médias (menores que $1Mb$) o custo do processo de decriptografia no

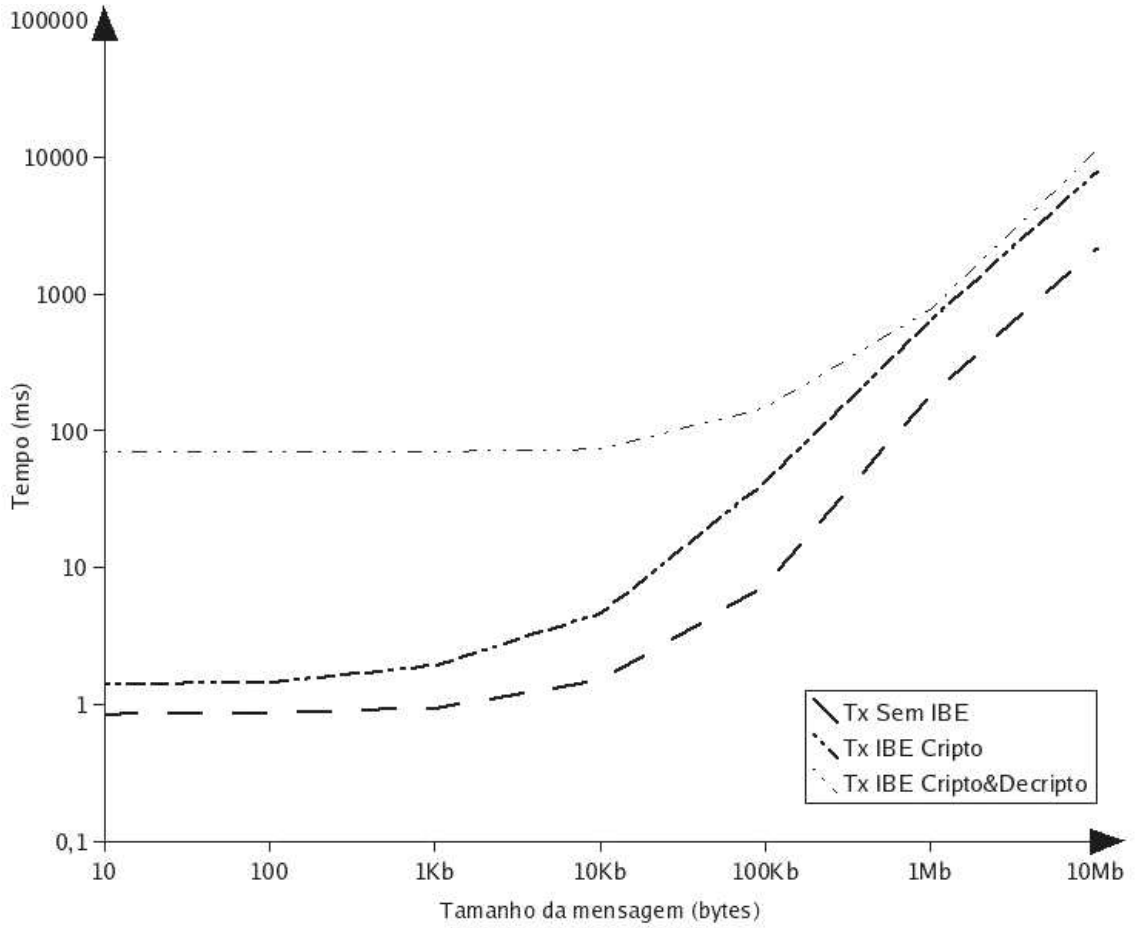

Figura 5.10: Comparativo para transmissão SMTP com e sem IBE

servidor causa um sensível impacto no tempo de transmissão no cliente. Esse "efeito" tende a diminuir quando são processadas mensagens grandes ($\geqslant 1Mb$).

Analisando a Figura 5.11, verifica-se quanto o tempo médio de conexão do servidor pode ser afetado ao se adotar criptografia no cliente e, opcionalmente, decriptografia no servidor. Nesse gráfico, o crescimento "linear" ficou por conta das linhas que representam conexões com ou sem decriptografia no servidor. Foi contabilizado um aumento médio de 1,44 vezes no tempo de conexão quando adotou-se o servidor sem e com decriptografia.

Prosseguindo a análise dos resultados apresentados pela Figura 5.11, observou-se que os tempos de conexão sem criptografia para mensagens pequenas (inferiores a $10Kb$) foi consideravelmente maior em cenários seguros. Em contrapartida,

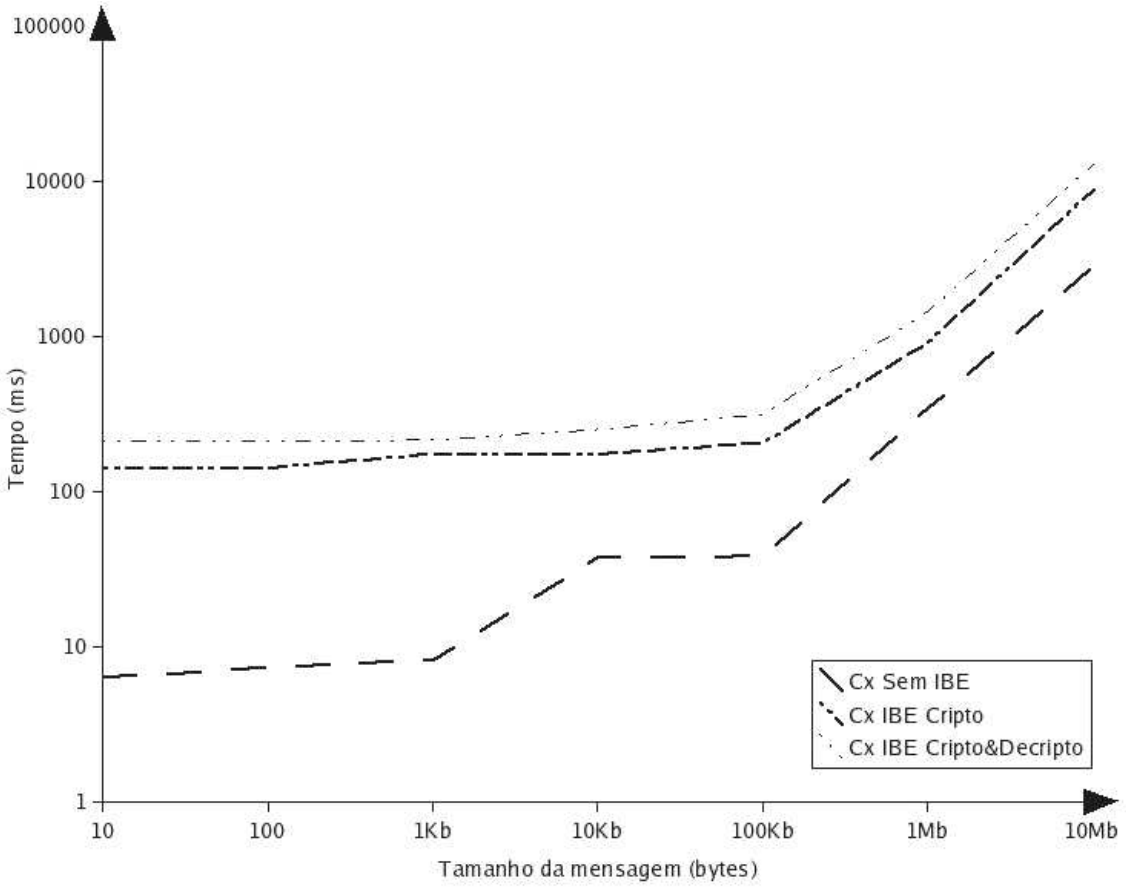

Figura 5.11: Comparativo para conexão SMTP com e sem IBE

mensagens com tamanho superior a $100Kb$ apresentaram um crescimento linear de, aproximadamente, $3,6$ vezes quando foi utilizada, por exemplo, somente criptografia no cliente. Esse valor é ligeiramente superior ao aumento (praticamente constante) de 3 vezes no tempo de transmissão nessa mesma situação, conforme indicado pela Figura 5.10. Logo, conclui-se que, para mensagens superiores a $100Kb$, o tempo de conexão fica praticamente limitado ao gasto com a transmissão dos dados cifrados.

Com o intuito de isolar as comparações entre os modelos com e sem criptografia implementada no servidor, foram inseridos os tempos de transmissão e conexão medidos entre execuções dessas variações de servidores em um único gráfico. Dessa forma, a Figura 5.12 compara as medições entre execuções de servidores com e sem decriptografia na recepção dessas mensagens. Através dessa figura, pode-se verificar o custo de adotar ou não decriptografia na implementação do servidor SMTP com

extensão IBE.

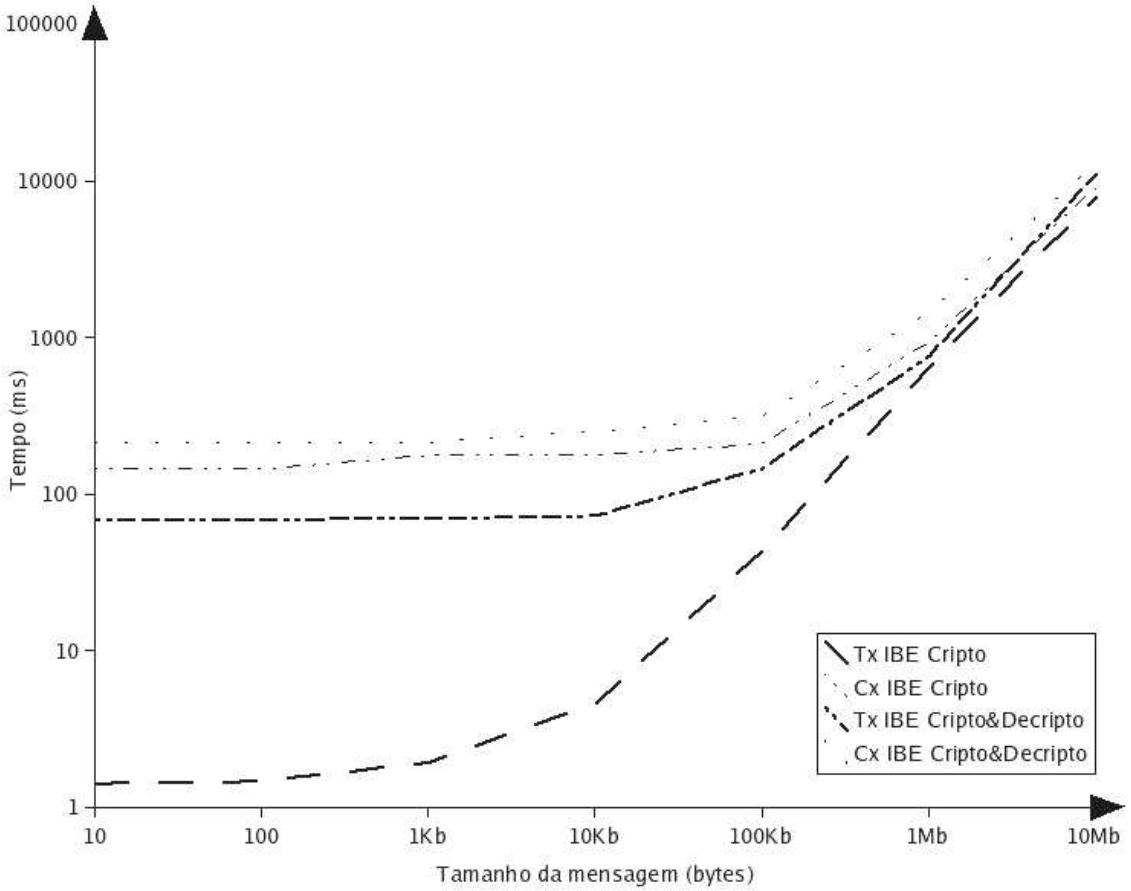

Figura 5.12: Comparativo entre servidores SMTP com extensão IBE

Uma das importantes informações passadas pela Figura 5.12 é o fato do tempo de transmissão ser sensivelmente afetado para transferências inferiores a $100Kb$, caso o servidor esteja decriptografando essa mensagem na sua recepção. De fato, seria bastante razoável esperar um resultado desses, uma vez que a maior carga no processamento de mensagens que chegam ao servidor acaba atrasando o consumo de informações por parte deste. Assim, acredita-se que o cliente seja "atrasado" devido a menor taxa de consumo dos dados disponibilizados para o servidor SMTP. Nesse sentido, testes utilizando ferramentas para medição de consumo do sistema (UCP, memória, etc), aliados a suspensões artificiais da execução do processo servidor durante o processo de transmissão, demonstraram que, embora essas tarefas estivessem executando na mesma máquina (conforme justificado na Tabela 5.3), não se pode atribuir tal atraso no tempo de transmissão do cliente ao custo associado ao compar-

tilhamento do sistema entre esses processos. Ficou latente o quanto a transmissão pode ser afetada ao se reduzir a taxa de consumo dos dados disponibilizados pelo cliente numa conexão SMTP.

Ainda na Figura 5.12, observa-se também que, para transmissões de dados superiores a $100Kb$, o próprio atraso na geração de uma maior massa de dados criptografados (por parte do cliente) faz com que o impacto no aumento do tamanho dessas mensagens não afete tanto esse tempo de transmissão, quando comparados a transmissões menores. Em outras palavras, para uma maior quantidade de dados, tempos de transmissão e conexão tendem a se aproximar. Isso também pode ser confirmado diretamente por essa figura, mesmo quando não há decriptografia no servidor.

Num balanço geral dos resultados vivenciados durante essa avaliação, verificou-se que o ônus gerado pelo processo de criptografia, claramente demonstrado pelo impacto visualizado nos tempos de execução, é totalmente compatível com o benefício trazido pela transmissão segura de mensagens, principalmente considerando essa ser uma implementação que ainda possui uma boa margem para melhorias. A questão seria, conseqüentemente, verificar se, ao trazer essas funcionalidades para a camada de transporte dessas mensagens, não se estaria comprometendo a eficiência dos serviços básicos oferecidos nessa camada. Com este objetivo, o Capítulo 6 irá, entre outros aspectos, tratar de algumas opções que constituem esforços no sentido de otimizar o desempenho do trabalho de implementação aqui proposto.

Capítulo 6

Conclusões e Trabalhos Futuros

ESTE capítulo tem o objetivo de não somente realizar uma avaliação final dos temas abordados e do trabalho realizado com base nesses temas, mas também levantar algumas melhorias e eventuais refinamentos em prosseguimento ao que foi feito. Assim, estar-se-á dividindo essa abordagem final em duas principais partes: Uma primeira parte destinada às conclusões, que pretende refletir sobre os benefícios e potencialidade relacionada ao advento das técnicas discutidas e, mais especificamente, sobre a proposta de emprego modelada e implementada aqui, e, finalmente, uma última parte contendo algumas sugestões para trabalhos futuros.

6.1 Conclusões

A conclusão desse trabalho também se dará de forma independente. Assim, serão abordadas, separadamente, a fase composta pelo estudo das técnicas apresentadas e, então, será realizada uma avaliação final sobre a proposta de modelo e implementação que acompanha esse texto.

6.1.1 Estudo e Potencial das Técnicas Apresentadas

Esse trabalho, ao apresentar um estudo sobre uma proposta na área de segurança, acabou por realizar um incentivo à pesquisa de soluções aplicando criptografia com curvas elípticas e criptografia por identidades. Dessa forma, foi possível observar os inquestionáveis benefícios associados à segurança e desempenho trazidos pela utilização de curvas elípticas em criptografia de chave pública. Ficaram também evidentes as vantagens no emprego de criptografia por identidades e todo o potencial e variação de cenários e problemas aos quais essa técnica pode responder. Finalmente, observou-se como esses dois "mundos" convergem para uma única e plenamente factível solução, cuja utilidade e aplicabilidade têm mostrado significativa abrangência.

Por se tratar de um assunto relativamente recente (boa parte da bibliografia citada é posterior a proposta de [3], de 2001) ainda há muito o quê ser descoberto e desenvolvido em termos de criptografia com curvas elípticas (e hiperelípticas), criptografia baseada em identidades e emparelhamentos. Dessa forma, muitos outros empregos, nas mais diversas áreas, certamente ainda poderão valer-se dessas técnicas.

6.1.2 Proposta de Extensão SMTP Apresentada

Os itens abaixo apresentam alguns dos principais aspectos que podem ser considerados dentro da idéia de solução para troca de mensagens segura contida nesse trabalho.

- Baseia-se em criptografia por identidades e, conseqüentemente, beneficia-se de todas as vantagens advindas dessa técnica, como melhora e simplificação do tradicional modelo de certificação e gerência de certificados.
- Devido à simplicidade da infra-estrutura citada acima, pôde ser inserido no contexto de uma extensão ao SMTP, sem grandes complexidades ou aumento expressivo do protocolo original.

- Constitui um nível a mais de segurança, adicionado dentro de um protocolo já existente e que já possui larga aceitação.
- A utilização ou não das funções de transferências seguras incorporadas ao SMTP não impede e tampouco "concorre" com essas funções implementadas no nível puramente aplicativo. Assim, soluções como *PGP* ou *S/MIME* podem conviver plenamente com essa proposta de SMTP seguro.
- Embora, a princípio, pudesse representar uma grande sobrecarga do protocolo de transporte, foi verificado que as conseqüências em termos de desempenho não chegam a causar nenhum impacto significativo, principalmente considerando os benefícios trazidos pela adoção dessa extensão.
- Não depende de nenhuma infra-estrutura "externa" ainda não existente, valendo-se do próprio protocolo SMTP para troca de parâmetros e codificação de mensagens seguras.
- De fácil utilização e implantação, tanto em clientes quanto em servidores SMTP, como ferramenta para envio de mensagens sigilosas.
- Apresentou total viabilidade de implementação partindo, inclusive, da adaptação de uma implementação real de cliente-servidor SMTP.

6.2 Trabalhos Futuros

Como complemento e trabalhos futuros, é possível listar algumas opções envolvendo melhorias aplicáveis à proposta de extensão SMTP apresentada. Nesse sentido, pode-se enunciar:

- Viabilização de um serviço compatível a esse no ambiente X.400, bem como atualização de *gateways* entre esses sistemas e o SMTP;
- Proposta de um padrão detalhando as estruturas de dados e parâmetros IBE trocados durante a conexão, de forma a desvincular essa solução ao utilizado pela biblioteca IBE [48] e poder inserir assinatura nesses dados;

- Verificação da validade dos parâmetros IBE de um servidor, produzindo uma nova versão de cliente que implemente validação de assinatura através da identidade do servidor SMTP, conforme o que foi tratado na Seção 5.2.6, bem como realização de avaliações de desempenho similares às mostradas na Seção 5.3.3;

- Detalhamento do formato de saída de mensagens criptografadas (versão *MTA* sem GCP), de forma a viabilizar implementações genéricas de leitores de *Mailbox* nos agentes usuários (*MUAs* destinatários);

- Idealização de um complemento à extensão SMTP proposta, visando inserir também garantias de autenticidade e integridade, em prosseguimento às idéias de parametrização local nos clientes SMTP e adoção de múltiplos GCPs, também observadas na Seção 5.2.6;

- Produção de uma biblioteca IBE inteiramente nova e com otimizações de desempenho;

- Estudo e correções eventualmente associadas a robustez e resistência a ataques na versão de servidor SMTP produzida;

- Melhorias de desempenho na versão de servidor *qmail* alterada, principalmente no relativo à manutenção de estruturas e arquivos temporários.

Bibliografia

[1] VOLTAGE. Email Security: The IBE Advantage. *Voltage Security Inc.* (July 2004).

[2] SHAMIR, A. Identity-Based Cryptosystems and Signature Schemes. In *Advances in Cryptology, Crypto 1984, Lecture Notes in Computer Science* (1984), Springs Verlag, pp. 47–53.

[3] BONETH, D., AND FRANKLIN, M. Identity-Based Encryption from The Weil Pairing. In *Crypto 2001* (2001).

[4] Voltage Security. http://www.voltage.com - Acesso em: 29/06/2004.

[5] MILLER, V. Uses of Elliptic Curves in Cryptography. In *Advances in Cryptology, Crypto 1985* (1986), Springs Verlag, pp. 417–426.

[6] KOBLITZ, M. Elliptic Curve Cryptosystems. In *Math Comp* (1987), pp. 203–209.

[7] POSTEL, J. Simple Mail Transfer Protocol. *RFC 821* (August 1982).

[8] KLESIN, J. Simple Mail Transfer Protocol. *RFC 2821* (April 2001).

[9] KLESIN, J., FREED, N., ROSE, M., STEFFERUD, E., AND CROCKER, D. SMTP Service Extensions. *RFC 1869* (November 1995).

[10] CERTICOM. Current Public-Key Cryptographic Systems. *A Certicom Whitepaper* (April 1997).

[11] COPPERSMITH, D. The Data Encryption Standard and Its Strength Against Attacks. *IBM Research Journal 38* (May 1994).

[12] DAEMEN, J., AND RIJMEN, V. AES Proposal: Rijndael. *AES Proposal* (June 1998).

[13] KOHL, J., AND NEUMAN, C. The Kerberos Network Authentication Service. *RFC 1510* (September 1993).

[14] DIFFIE, W., OORSCHOT, P. V., AND WIENER, M. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography 2* (1992), 107–125.

[15] DIFFIE, W., AND HELLMAN, M. L. New Directions in Cryptography. *IEEE Transactions on Information Theory 22* (1976), 644–654.

[16] SCHROEPPEL, R., ORMAN, H., AND MALLEY, S. O. Fast Key Exchange with Elliptic Curve Systems. *Departament of Computer Science - University of Arizona* (March 1995).

[17] STALLINGS, W. *Cryptography and Network Security - Principles and Practice*, 2nd ed. Prentice Hall.

[18] CERTICOM. An Introduction to Information Security. *A Certicom Whitepaper* (March 1997).

[19] RAUSCHER, R., AND BOHNSACK, F. Results of na Elliptic-Curve-Approach for Use in Cryptosystems. In *EUROMICRO Conference* (1999), pp. 415–422.

[20] CERTICOM. Remarks on the Security of the Elliptic Curve Cryptosystem. *A Certicom Whitepaper* (September 1997).

[21] ELGAMAL, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory 31* (1985), 469–472.

[22] KOBLITZ, N., MENEZES, A., AND VANSTONE, S. The state of elliptic curve cryptography. *Designs, Codes and Cryptography* (2000), 173–193.

[23] MENEZES, A. Elliptic Curve Public Key Cryptosystems. *Kluwer Academic Publishers* (2001).

[24] RABIN, M. O. Digitalized Signatures and Public-Key Functions as Intractable as Factorization. *MIT Laboratory for Computer Science* (1979).

[25] KALISKI, B., AND STADDON, J. RSA Cryptography Specifications Version 2.0. *RFC 2437* (October 1998).

[26] Digital Signature Standard. In *FIPS Publication 186* (1993). National Institute of Standards and Technology.

[27] GUPTA, V., STEBILA, D., FUNG, S., SHANTZ, S. C., GURA, N., AND EBERLE, H. Speeding up Secure Web Transactions using Elliptic Curve Cryptography. In *Network and Distributed System Security Symposium* (2004).

[28] KALISKY, B. A Pseudorandom Bit Generator Based on Elliptic Logarithms. In *Advances in Cryptology, Crypto 1986* (1987), vol. 293, Springs Verlag, pp. 84–103.

[29] GOLDWASSER, S., AND KILIAN, J. Almost All Primes can be Quickly Certified. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing* (1986), pp. 316–329.

[30] BOTES, J., AND PENZHORN, W. Public-Key Cryptosystems Based on Elliptic Curves. In *Communications and Signal Processing Proceedings of the 1993 IEEE South African Symposium* (August 1993).

[31] BELINGUERES, G. Introduccion A Los Criptosistemas de Curva Eliptica. http://www.toptutoriales.com/matematicas/criptografia - Acesso em: 30/12/2002.

[32] WOLSKI, E. Sistemas Criptográficos Baseados em Curvas Elípticas. http://www.unb.br - Acesso em: 10/01/2003.

[33] ELSAYED MOHAMMED, A. E. E., AND EL-SHENNAWY, K. Elliptic Curve Cryptosystems on Smart Cards. In *IEEE 35th International Carnahan Conference* (October 2001), pp. 213–222.

[34] ECC On Line Tutorial. http://www.certicom.com - Acesso em: 17/10/2002.

[35] Elliptic Curve Cryptography FAQ. www.cryptoman.com/elliptic.htm - Acesso em: 15/10/2002, December 1997.

[36] CHOUINARD, J. Y. Notes on Elliptic Curve Cryptography - Design of Secure Computer Systems. http::/www.site.uottawa.ca - Acesso em: 15/12/2002, September 2002.

[37] DESMEDT, Y., AND QUISQUATER, J. Public-Key Systems Based on The Difficulty of Tampering. In *Advances in Cryptology, Crypto 1986* (1986), vol. 22, Springs Verlag, pp. 111–117.

[38] HUHNLEIN, D., JACOBSON, M., AND WEBER, D. Towards Practical Non-interactive Public Key Cryptosystems Using Non-maximal Imaginary Quadratic Orders. In *Selected Areas in Cryptography* (2000), vol. 2012, Springs Verlag, pp. 275–287.

[39] MAURER, U., AND YACOBI, Y. Non-interactive Public-key Cryptography. In *Advances in Cryptology, Crypto 91* (1991), vol. 547, Springs Verlag, pp. 498–507.

[40] TSUJI, S., AND ITOH, T. An ID-based Cryptosystem Based on the Discrete Logarithm Problem. *IEEE Journal on Selected Areas in Communication 7* (1989), 467–473.

[41] TANAKA, H. A Realization Scheme for Identity-Based Cryptosystem. In *Advances in Cryptology, Crypto 87* (1987), vol. 293, Springs Verlag, pp. 341–349.

[42] SHAMIR, A., FIAT, A., AND FEIGE, U. Zero-knowledge Proofs of Identity. *J. Cryptology 1* (1988), 77–94.

[43] SHAMIR, A., AND FEIGE, U. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology, Crypto 86* (1986), vol. 263, Springs Verlag, pp. 186–194.

[44] GALBRAITH, S., HARRISON, K., AND SOLDERA, D. Implementing the Tate Pairing. In *Lecture Notes in Computer Science* (2002), vol. 2369, Springs Verlag, pp. 324–337.

[45] Barreto, P. Efficient Algorithms for Pairing-based Cryptosystems. In *Advances in Cryptology, Crypto 2002* (2002), vol. 2442, Springs Verlag, pp. 354–368.

[46] Barreto, P., Lynn, B., and Scott, M. On The Selection of Pairing-Friendly Groups. http://eprint.iacr.org/2003/086 - Acesso em 20/12/2004, 2003.

[47] Junior, W. D. B. Sistemas Criptográficos Baseados em Identidades Pessoais. Tese de Mestrado, Universidade de São Paulo, Instituto de Matemática e Estatística, Novembro 2003.

[48] Lynn, B. IBE Secure E-mail. http://crypto.stanford.edu/ibe - Acesso em: 13/07/2004.

[49] Chen, L., Harrison, K., Soldera, D., and Smart, N. P. Applications of Multiple Trust Authorities in Pairing Based Cryptosystems. In *Proceedings of InfraSec 2002* (2002), vol. 2437, Springs Verlag, pp. 260–275.

[50] Chen, X., Zhang, F., and Kim, K. ID-based Multi-Proxy Signature and Blind Multisignature from Bilinear Pairings. In *Proceedings of KIISC'2003* (August 2003), pp. 11–19.

[51] Duursma, I., and Lee, H. Tate Pairing Implementation for Hyperelliptic Curves. In *Advances in Cryptology, Asiacrypt'2003, Lecture Notes in Computer Science 2894* (2003), Springs Verlag, pp. 111–113.

[52] Barreto, P. S. L. M. *Criptografia Robusta e Marcas d'Água Frágeis: Construção e Análise de Algoritmos para Localizar Alterações em Imagens Digitais.* Tese de Doutorado, Universidade de São Paulo, Escola Politécnica, 2003.

[53] Mont, M. C., Harrison, K., and Sadler, M. The HP Time Vault Service. http://www.hpl.hp.com/techreports - Acesso em: 23/06/2004, September 2002.

[54] Mont, M. C., Bramhall, P., and Harrison, K. A flexible role-based secure messaging service: Exploiting IBE Technology for Privacy in Helth

Care. http://www.hpl.hp.com/techreports - Acesso em: 20/01/2005, September 2003.

[55] Mu, Y., and Varadharajan, V. An Efficient Internet Credit Card Scheme from the Weil Pairing. IEEE Press, pp. 58–63.

[56] Hess, F. Efficient Identity-Based Signature Schemes Based on Pairings. In *Lecture Notes in Computer Science* (2002), vol. 2595, Springs Verlag, pp. 310–324.

[57] Cha, J. C., and Cheon, J. H. An Identity-Based Signature from Gap Diffie-Hellman Groups. In *Practice and Theory in Public Key Cryptography - PKC'2003* (2003), Springs Verlag, pp. 18–30.

[58] Herranz, J., and Sáez, G. A Provably Secure ID-based Ring Signature Scheme. http://eprint.iacr.org/2003/261 - Acesso em 15/01/2005.

[59] Heb, F. Efficient Identity Based Signature Schemes Based on Pairings. In *Selected Areas in Cryptography - SAC'2002, Lecture Notes on Computer Science* (2002), Springs Verlag, pp. 310–324.

[60] Hu, F., Wu, C., and Irwin, J. D. A New Forward Secure Signature Scheme using Bilinear Maps. http://eprint.iacr.org/2003/188 - Acesso em 18/12/2004.

[61] Kurosawa, K., and Heng, S. H. From Digital Signature to ID-Based Identification / Signature. In *Practice and Theory in Public Key Cryptography - PKC'2004* (March 2004), Springs Verlag, pp. 248–261.

[62] Libert, B., and Quisquater, J. J. Identity Based Undeniable Signatures. http://eprint.iacr.org/2003/206 - Acesso em: 17/12/2004.

[63] Paterson, K. G. ID-based Signatures from Pairings on Elliptic Curves. 2002.

[64] Smart, N. P. An Identity Based Authenticated Key Agreement Protocol Based on the Weil Pairing. *Electronics Letters 38* (2002), 630–632.

[65] McCullagh, N., and Barreto, P. S. L. M. A New Two-Party Identity-Based Authenticated Key Agreement. http://eprint.iacr.org/2004/122 - Acesso em 10/01/2005.

[66] Chen, L., and Kudla, C. Identity Based Authenticated Key Agreement from Pairings. http://eprint.iacr.org/2002/184 - 06/01/2005.

[67] Choi, K. Y., Hwang, J. Y., and Lee, D. H. Efficient ID-based Group Key Agreement with Bilinear Maps. In *Practice and Theory in Public Key Cryptography - PKC'2004* (March 2004), Springs Verlag, pp. 130–144.

[68] X. Du, Y. W., and Ge, J. ID-based Authenticated Two Round Multi-Party Key Agreement. http://eprint.iacr.org/2003/247 - Acesso em: 10/01/2005.

[69] Liu, S., Zhang, F., and Chen, K. ID-Based Tripartite Key Agreement Protocol with Pairings. In *Proceedings of the 2003 IEEE International Symposium on Information Theory* (2003), pp. 136–136.

[70] Sowers, J. A. Identity-Based Encryption to Eliminate Certificates in SSL Transactions. March 2002.

[71] Zhang, F., and Kim, K. Signature-Masked Authentication Using the Bilinear Pairings. Cryptology and Information Security Laboratory, September 2002.

[72] Baldwin, M. Identity Based Encryption from the Tate Pairing to Secure Email Communications. Tese de Mestrado, University of Bristol, Bristol, UK, May 2002.

[73] Barreto, P. The Pairing-Based Crypto Lounge. http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html - Acesso em: 15/01/2005.

[74] Atkins, D., Stallings, W., and Zimmermann, P. PGP Message Exchange Formats. *RFC 1991* (August 1996).

[75] Dusse, S., Hoffman, P., Ramsdell, B., Lundblade, L., and Repka, L. S/MIME Version 2 Message Specification. *RFC 2311* (March 1998).

[76] Myers, and Rose, M. Post Office Protocol - Version 3. *RFC 1939* (May 1996).

[77] Crispin, M. Internet Message Access Protocol - Version 4rev1. *RFC 3501* (March 2003).

[78] International Organization for Standardization. http://www.iso.ch - Acesso em: 10/01/2004.

[79] International Telecomunication Union. http://www.itu.ch - Acesso em: 12/01/2004.

[80] Rhoton, J. *X.400 and Smtp: Battle of the E-Mail Protocols*, 2nd ed. Digital Press.

[81] Kille, S. Mapping between X.400 and RFC 822. *RFC 987* (June 1986).

[82] Crocker, E. D., and Overell, P. Augmented BNF Syntax Specifications: ABNF. *RFC 2234* (November 1997).

[83] Hoffman, P. SMTP Service Extension for Secure SMTP over TLS. *RFC 2487* (January 1999).

[84] Myers, J. SMTP Service Extension for Authentication. *RFC 2554* (March 1999).

[85] Bradner, S. Key words for use in RFCs to Indicate Requirement Levels. *RFC 2119* (March 1997).

[86] Bradner, S. TELNET Protocol Specification. *RFC 854* (May 1983).

[87] McClure, S., Scambray, J., and Kurtz, G. *Hacking Exposed: Network Security Secrets & Solutions*, 4th ed. Computing Mcgraw-Hill.

[88] Bernstein, D. J. qmail. http://cr.yp.to/qmail.html - Acesso em: 27/11/2004.

[89] GNU Multiple Precision Arithmetic Library. http://www.swox.com/gmp - Acesso em: 12/07/2004.

[90] OpenSSL Project. http://www.openssl.org - Acesso em: 12/07/2004.

[91] Menezes, A. J., Oorshot, P. C. V., and Vanstone, S. A. *Handbook of Applied Cryptography*, 5th ed. CRC Press.

Apêndice A

Álgebra Abstrata

Este apêndice apresenta algumas definições básicas relacionadas a álgebra abstrata, bem como seus objetos e propriedades associadas aos mesmos. Essas e outras definições e informações complementares podem ser encontradas em [31][91].

Definição A.1 Uma *operação binária* $*$ em um conjunto S é um mapeamento $S \times S \longrightarrow S$. Dessa forma, $*$ é uma regra que associa a cada par ordenado de elementos em S, um elemento em S.

Definição A.2 Um *grupo* $(G, *)$ consiste em um conjunto G com uma operação binária $*$ em G, satisfazendo aos seguintes axiomas:

1. A operação de grupo é *associativa*. Logo, $a * (b * c) = (a * b) * c$, para todos a, b, $c \in G$.
2. Existe um elemento $1 \in G$, chamado de *elemento identidade*, onde $a * 1 = 1 * a = a$ para todo $a \in G$.
3. Para cada $a \in G$ existe um elemento $a^{-1} \in G$, chamado de *inverso* de a, de forma que $a * a^{-1} = a^{-1} * a = 1$.
4. Um grupo G é dito *grupo abeliano* (ou comutativo) se: $a * b = b * a$ para todos a, $b \in G$.

Pode-se notar que a notação de *grupo multiplicativo* foi utilizada nessa definição. Normalmente, a operação $*$ em grupos multiplicativos é omitida e, dessa

forma, $a * b$ é representado também por ab, onde a, $b \in G$.

Quando a operação de grupo é adição, então o grupo é chamado de *grupo aditivo*, o elemento identidade passa a ser 0 e o inverso de a passa a ser $-a$.

Definição A.3 Um grupo G é *finito* se $|G|$ é finito. O número de elementos desse grupo finito é chamado de *ordem* desse grupo.

Definição A.4 Um grupo G é *cíclico* se existe um elemento $\alpha \in G$ de forma que para cada elemento $b \in G$, existe um inteiro i tal que $b = \alpha^i$. Esse elemento α é chamado de *gerador* de G.

Fato A.5 Se G é um grupo e $a \in G$, então o conjunto de todas as potências de a formam um subgrupo cíclico de G, chamado de subgrupo *gerado por* a, sendo denotado por $\langle a \rangle$.

Definição A.6 Seja G um grupo e $a \in G$. A ordem de a é o menor inteiro positivo t tal que $a^t = 1$, caso t exista. Se t não existe, então a ordem de a é ∞.

Fato A.7 Se G é um grupo e $a \in G$ é um elemento de ordem t, então $|\langle a \rangle| = t$ é o tamanho do subgrupo gerado por a.

Fato A.8 Todo subgrupo de um grupo cíclico G também é cíclico. Se G é um grupo cíclico de ordem n, então para cada divisor positivo d de n, G contém exatamente um subgrupo de ordem d.

Exemplo Considerando o grupo multiplicativo $\mathbb{Z}_{19}^* = \{1,\ 2, \ldots\ 18\}$ de ordem 18. Esse grupo é cíclico e seu gerador é $\alpha = 2$. Os subgrupos de $\mathbb{Z}_{19}^*$ e seus respectivos geradores podem ser vistos na Tabela A.1.

Definição A.9 Um *anel* $(R,\ +,\ \times)$ consiste em um conjunto R com duas operações binárias arbitrariamente definidas $+$ (adição) e $\times$ (multiplicação) em R, satisfazendo aos seguintes axiomas:

1. $(R,\ +)$ é um grupo abeliano com identidade 0.
2. A operação $\times$ é associativa, logo: $a \times (b \times c) = (a \times b) \times c$, para todos a, b, $c \in R$.

Subgrupo	Geradores	Ordem
{1}	1	1
{1, 18}	18	2
{1, 7, 11}	7, 11	3
{1, 7, 8, 11, 12, 18}	8, 12	6
{1, 4, 5, 6, 7, 9, 11, 16, 17}	4, 5, 6, 9, 16, 17	9
{1, 2, 3, ..., 18}	2, 3, 10, 13, 14, 15	18

Tabela A.1: Os subgrupos de $\mathbb{Z}_{19}^{*}$ [91]

3. Existe uma identidade multiplicativa 1, $1 \neq 0$, de forma que $1 \times a = a \times 1 = a$, para todo $a \in R$.

4. A operação $\times$ é distributiva em $+$, logo: $a \times (b + c) = (a \times b) + (a \times c)$, para todos a, b, $c \in R$.

Definição A.10 Um anel é um *anel comutativo* se $a \times b = b \times a$, para todos a, $b \in R$.

Definição A.11 Um elemento a de um anel R é chamado de *elemento inversível* se existe algum elemento $b \in R$ tal que: $a \times b = 1$.

Apêndice B

Corpos Finitos

Este apêndice apresenta algumas definições básicas relacionadas a corpos e, em especial, corpos finitos. Algumas das definições utilizadas aqui podem ser encontradas no Apêndice A.

Definição B.1 Um *corpo* $\mathbb{F}$ é um anel comutativo no qual todos os elementos diferentes de 0 têm inversos (Definições A.10 e A.11).

Definição B.2 A *característica* de um corpo $\mathbb{F}$ é o menor inteiro positivo m tal que $\sum_{i=1}^{m} 1 = 0$. A característica de um corpo é 0 se $\overbrace{1+1+...+1}^{m}$ nunca é 0 para qualquer $m \geq 1$.

Fato B.3 $\mathbb{Z}_n$ é um corpo a partir de operações de adição e multiplicação em módulo n, se e somente se n é um número primo. Se n é primo, então $\mathbb{Z}_n$ tem característica n.

Definição B.4 Um *corpo finito* é um corpo $\mathbb{F}$ que contém um número finito de elementos.

Definição B.5 A *ordem* de um corpo finito $\mathbb{F}$ é o número de elementos em $\mathbb{F}$. Um corpo finito de ordem q é denotado por $\mathbb{F}_q$.

Definição B.6 Os elementos diferentes de 0 em $\mathbb{F}_q$ formam um grupo multiplicativo chamado de *grupo multiplicativo* de $\mathbb{F}_q$, denotado por $\mathbb{F}_q^*$.

Fato B.7 $\mathbb{F}_q^*$ é um grupo cíclico de ordem q - 1. Logo: $a^q = a$, para todo $a \in \mathbb{F}_q$.

Definição B.8 O gerador do grupo cíclico $\mathbb{F}_q^*$ é chamado de *elemento primitivo* ou *gerador* de $\mathbb{F}_q$.

Fato B.9 Se a, $b \in \mathbb{F}_q$ (um corpo finito de característica p), então:

$$(a+b)^{p^t} = a^{p^t} + b^{p^t}, \text{ para todo } t \geq 0.$$