UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Um Modelo de Infra-Estrutura de Chaves Públicas para um Correio Eletrônico Seguro

por

MÁRCIA PEDRINI

Trabalho de Conclusão de Mestrado submetido à avaliação,
como requisito parcial para obtenção do grau de Mestre
em Informática

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, maio de 2004

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Pedrini, Márcia

Um Modelo de Infra-Estrutura de Chaves Públicas para um Correio Eletrônico Seguro / por Márcia Pedrini. – Porto Alegre:PPGC da UFRGS, 2004.

151 f.:il.

Trabalho de Conclusão (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2004. Orientador: Geyer, Cláudio Fernando Resin.

1. Certificação Digital. 2. Infra-estrutura de chaves públicas. 3. Segurança da Informação. 4. Direto. I. Geyer, Cláudio Fernando Resin. II. Título.

Agradecimentos

Agradeço ao Prof. Cláudio Fernando Resin Geyer, que muito mais que um orientador, foi um apoiador, um incentivador e, quando tudo parecia perdido, não me deixou desanimar.

Meu muito obrigada ao Prof. Jeroen Antonios Maria Van de Graaf, que utilizou o feriado do Carnaval do ano de 2004 para ler o meu trabalho, exatamente na época em que eu o terminava e surgiam as dúvidas se ele estava completo ou faltava alguma coisa.

Ao Sr. Carlos Zen, que me apoiou na decisão de cursar um mestrado e fez tudo que estava ao seu alcance para permitir a minha entrada no curso.

Aos meus irmãos Mauro e Mairo, que foram os melhores irmãos - teóricos e práticos - que um mestrando pode ter.

À toda a minha família, que me deu o apoio, o carinho, a preocupação e também a cobrança necessária para a realização deste trabalho.

Aos meus amigos e amigas, que sempre acreditaram que eu ia conseguir.

A todos aqueles que não me deixaram esquecer, por um segundo sequer, que é necessário ser perseverante para se conquistar um sonho.

Sumário

Lista de Abreviaturas **7**

Lista de Figuras **9**

Lista de Tabelas **11**

Resumo **12**

Abstract **13**

1 Introdução **14**

1.1 Motivação **14**
1.2 Descrição dos Capítulos **16**

2 Infra-estrutura de chaves públicas **18**

2.1 Definição **18**
2.2 Componentes **18**
2.2.1 Entidades Finais 19
2.2.2 Autoridade Certificadora 19
2.2.3 Autoridade Registradora 20
2.2.4 Repositório 20
2.2.5 Chaves 21
2.2.6 Assinatura Digital 22
2.2.7 Certificados 24
2.2.8 Lista de Certificados Revogados 25
2.3 Serviços **26**
2.4 Características e Funcionalidades **26**
2.5 Arquitetura **28**
2.6 Ciclo de Vida de um certificado **29**
2.6.1 Emissão 29
2.6.2 Suspensão 30
2.6.3 Revogação 30
2.6.4 Expiração 31
2.6.5 Renovação 32
2.7 Documentos de Política **32**
2.7.1 Política de certificados (PC) 32
2.7.2 Declaração de Práticas de Certificação (DPC) 33
2.8 Administração de uma ICP **35**
2.9 Considerações Finais **35**

3 Ferramentas, Tecnologias, Representação de Dados e Padrões **36**

3.1 Criptografia **36**

3.1.1 Criptografia Convencional 36
3.1.2 Criptografia de Chave Pública 37
3.1.3 Hash Criptográfico 37
3.2 LDAP 38
3.2.1 Modelos LDAP 39
3.2.2 Características do LDAP 40
3.2.3 Implantando um serviço de diretórios distribuídos 40
3.3 OpenLDAP 44
3.4 OpenCA 44
3.5 OpenSSL 47
3.6 Pacote JDK 1.4.2 da Sun Microsystems 47
3.7 KeyTool 48
3.8 Bouncy Castle 50
3.9 Representação de Dados 51
3.9.1 ASN.1 51
3.9.2 BER 52
3.9.3 DER 52
3.10 Padrões 53
3.10.1 PKCS #8 - Private-Key Information Syntax Standard 53
3.10.2 PKCS #10 - Certification Request Syntax Standard 54
3.10.3 PKCS #5 - Password-Based Cryptography Standard 56
3.11 Considerações Finais 58

4 Direto 59

4.1 Projeto Direto 59
4.2 O produto Direto 61
4.2.1 Tecnologias Utilizadas 61
4.2.2 Arquitetura do Direto 63
4.3 Projeto Direto / Fapergs 64

5 Modelo Geral 66

5.1 Introdução 66
5.2 Avaliando a necessidade de uma ICP 66
5.3 Definindo o modelo de confiança 68
5.4 Definindo a necessidade de ARs 69
5.5 Certificados 70
5.5.1 Formato de Certificado 70
5.5.2 Estrutura dos Certificados 73
5.6 Definindo a ICP da Procergs 79
5.6.1 Funcionalidades da ICP-Pro 80
5.6.2 Autoridade Certificadora AC-Pro 86
5.6.3 Autoridade Registradora da Procergs (AR-Pro) 97
5.7 Definindo políticas de segurança 100
5.7.1 Declaração de Práticas de Certificação (DPC) 100
5.8 Interfaces 101
5.9 Considerações Finais 101

6 Protótipo da Infra-estrutura de Chaves Públicas 103

6.1 Considerações Iniciais 103
6.2 Armazenamento dos Dados 103
6.2.1 Arquivo de Configurações 103
6.2.2 LDAP 104
6.2.3 PostgreSQL 104
6.3 Criação da AC-Pro 106
6.3.1 Submenu AC 106
6.3.2 Submenu Usuário 112
6.4 Alteração do Direto 113
6.4.1 Gera Certificado 115
6.4.2 Renova Certificado 121
6.4.3 Revoga Certificado 121
6.5 Envio de e-mail 122
6.6 Recepção de e-mail 124
6.7 Considerações Finais 127

7 Teste e validação do protótipo 128

7.1 Geração de Certificado 128
7.2 Criptografia de mensagem 128
7.3 Cópia de e-mail 129
7.4 Assinatura da Mensagem 129
7.5 Considerações Finais 129

8 Trabalhos Futuros 130

8.1 Trabalhos futuros – curto prazo 130
8.1.1 Gerenciamento de múltiplos pares de chaves 130
8.1.2 Interface para solicitação de certificados 131
8.1.3 Cadeia de certificados 132
8.1.4 Revogação do certificado de uma das ACs 133
8.2 Trabalhos futuros – médio prazo 134
8.2.1 Divulgação de certificados revogados 134
8.2.2 Mobilidade 134
8.2.3 Armazenamento de chaves por dispositivos de hardware 135
8.2.4 Registro de Data / Hora 135
9 Conclusão 136

Referências 138

Anexo Declaração de Práticas de Certificação da Autoridade Certificadora da Procergs (DPC da AC-Pro) 144

Lista de Abreviaturas

AC	Autoridade Certificadora
AES	Advanced Encryption Standard
API	Application Program Interface
AR	Autoridade Registradora
ARPA	Advanced Research Projects Agency
ASN.1	Abstract syntax notation
BER	Basic Encoding Rules
BNF	Backus Naur Form
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CGI	Common Gateway Interface
CMS	Content Management System
CN	Common Name
CORBA	Common Object Request Broker Architecture
DAP	Directory Access Protocol
DER	Distinguished Enconding Rules
DES	Data Encryption Standard
DMZ	Demilitarized Zone
DN	Distinguished Name
DOM	Document Object Model
DPC	Declaração de Práticas de Certificação
EF	Entidade Final
FAPERGS	Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul
GPPD	Grupo de Processamento Paralelo e Distribuído
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ICP	Infra-estrutura de Chaves Públicas
IDEA	International Data Encryption Algorithm
II-UFRGS	Instituto de Informática – Unifersidade Federal do Rio Grande do Sul
IMAP	Internet Message Access Protocol
IP	Internet Protocol
ISO	International Standards Organization
ITU	International Telecommunication Union
J2SE	Java 2 Standard Edition
JCA	J2EE Connector Architecture
JCE	Java Cryptography Extension
JDBC	Java Database Connectivity
JDK	Java Development Kit
JNDI	Java Naming and Directory Interface
JSP	Java Server Pages
JSSE	Java Secure Socket Extension
KDF	Key Derivation Function
LCR	Lista de Certificados Revogados
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code

METROPLAN	Fundação Estadual do Planejamento Metropolitano e Regional
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PBE	Password-Based Encryption
PC	Política de Certificados
PCMCIA	Personal Computer Memory Card International Association
PDA	Personal Digital Assistants
PIN	Personal Identification Number
PKCS	Public-Key Cryptography Standards
PKIX	Public-Key Infrastructure X.509
PROCERGS	Companhia de Processamento de Dados do Rio Grande do Sul
RC	Repositório de Certificados
RDN	Relative Distinguished Name
RFC	Request for Comments
RSA	Rivest, Shamir e Adleman
S/MIME	Secure / Multipurpose Internet Mail Extensions
SERPRO	Serviço Federal de Processamento de Dados
SGML	Standard Generalized Markup Language
SHA	Secure Hash Algorithm
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
SUSEPE	Superintendência dos Serviços Penitenciários
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UCPel	Universidade Católica de Pelotas
URCAMP	Universidade da Região da Campanha
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WPKI	Wireless PKI
WTLS	Wireless Transport Layer Security
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language

Lista de Figuras

FIGURA 2.1 – Principais componentes de uma ICP 19
FIGURA 2.2 - Criação e verificação de uma assinatura digital 23
FIGURA 2.3 - Certificado 25
FIGURA 2.4 – Arquitetura física de uma ICP 28
FIGURA 2.5 – Ciclo de vida esperado de um certificado 29
FIGURA 3.1 – Acesso ao diretório 39
FIGURA 3.2 – Exemplo de Espaço de Nomes 41
FIGURA 3.3 – um modelo de topologia 42
FIGURA 3.4 – Acesso ao servidor, utilizando referral 43
FIGURA 3.5 – exemplo de interface no projeto OpenCA 46
FIGURA 3.6 – Estruturação do *keystore* 49
FIGURA 4.1 – Arquitetura do Direto 64
FIGURA 5.1 - Arquitetura do DiretoGNU 67
FIGURA 5.2 – Modelo de Confiança da ICP da PROCERGS 68
FIGURA 5.3 – Instalação das ARs 69
FIGURA 5.4 – Principais componentes da ICP-Pro 79
FIGURA 5.5 – Criptografia baseada em senha 81
FIGURA 5.6 – Decriptografia baseada em senha 82
FIGURA 5.7 - Estudo de Caso – Geração de Certificados 84
FIGURA 5.8 – Continuação do Estudo de Caso – Geração de Certificados 85
FIGURA 5.9 – Estudo de Caso da Revogação de Certificados 90
FIGURA 5.10 - modelo de utilização da LCR 92
FIGURA 5.11 – definição da tabela AR_solicitacoes 99
FIGURA 6.1 - exemplo de arquivo configurado pelo usuário 104
FIGURA 6.2 – Modelagem das tabelas do PostgreSQL 105
FIGURA 6.3 – Interface do módulo AC 106
FIGURA 6.4 – Submenu AC 106
FIGURA 6.5 – Código utilizado para a criação do par de chaves da AC-Pro 107
FIGURA 6.6 - Método para Criação do Certificado da AC Raiz 109
FIGURA 6.7 – O Certificado da AC-Pro 110
FIGURA 6.8 – Certificado da AC-Pro instalado no navegador. 111
FIGURA 6.9 - Configuração da AC-Pro 112
FIGURA 6.10 – Submenu Usuário 113
FIGURA 6.11 – Interface do correio do Direto 114
FIGURA 6.12 – Novo item adicionado ao Direto 115
FIGURA 6.13 – Termo de Aceitação do Certificado 116
FIGURA 6.14 – Tela de login do sistema Direto 117
FIGURA 6.15 – E-mail avisando sobre dados incompletos 118
FIGURA 6.16 – Criptografia e armazenamento da chave privada 119
FIGURA 6.17 – Certificado do usuário 120
FIGURA 6.18 – LCR da AC-Pro 122
FIGURA 6.19 – novas funcionalidades no envio de e-mail 123
FIGURA 6.20 – Interface de recebimento de e-mail do Direto 125
FIGURA 6.21 – Remetente não autenticado 126

FIGURA 8.1 – Um modelo de interface..... 131
FIGURA 8.2 – Cadeia de certificados..... 132

Lista de Tabelas

TABELA 5.1 – Definição dos campos do certificado 73
TABELA 5.2 – Valores válidos para *Key Usage* 75
TABELA 5.3 – Valores válidos de ReasonCode 76
TABELA 5.4 - Tipos possíveis de Nomes Alternativos 78
TABELA 5.5 – campos da LCR do Direto 94
TABELA 5.6 – crlEntryExtension para a LCR do Direto 95
TABELA 5.7 – Extensões da LCR 95
TABELA 5.8 – razões de revogação e seus prazos para divulgação 97

Resumo

Com o crescimento constante do uso do Correio Eletrônico, pela sua facilidade de uso e por sua rapidez na entrega da correspondência, cresce também a preocupação com a segurança no tráfego dos dados. Esta segurança é cada vez mais enfatizada, principalmente quando imagina-se a possibilidade de se ter as informações expostas a intrusos da Internet, que surgem com meios cada vez mais sofisticados para violar a privacidade e a segurança das informações.

A certificação digital e a infra-estrutura de chaves públicas formam juntas a arquitetura de segurança mais utilizada para garantir os quatro quesitos básicos de segurança em correio eletrônico, que são: autenticação, integridade, não-repúdio, privacidade e datação.

O Direto, desenvolvido pela PROCERGS, Companhia de Processamento de Dados do Estado do Rio Grande do Sul, tem como objetivo principal atender a demanda de um software de comunicação de baixo custo, que interligue os diversos órgãos do estado. Por suas características, o sistema está suscetível a potenciais falhas de segurança. O Direto é baseado em software livre, o que diminui o custo do projeto e possibilita o seu uso por qualquer tipo de organização interessada, sem custo algum, com independência de plataforma e flexibilidade de aperfeiçoamento.

O modelo proposto e o protótipo desenvolvido visam garantir a segurança das informações trafegadas pelo módulo de correio eletrônico do Direto, estendendo sua interface e incorporando os conceitos de certificados eletrônicos e infra-estrutura de chave pública, atendendo os quatro quesitos básicos de segurança.

Palavras-chave: Certificação Digital, Infra-estrutura de chaves públicas, Direto, segurança da informação.

TITLE: “AN INFRASTRUCTURE MODEL OF PUBLIC KEYS FOR A SECRET E-MAIL”

Abstract

With the growth of Eletronic Mail, caused by its ease of use and fast message deliver process, also grows the preocupation in security of transmitted data. This security is more and more emphasized, even more when there is a chance of having information exposed to intruders from the Internet that appears with ways more and more sophisticated for breaking the privacy and security of information.

Digital Certification and public-key infrastructure are the current security standards used to ensure the five basic itens of electronic mail security: authenticity, integrity, privacy, non-repudiation and timestamping.

Direto, developed by PROCERGS, Companhia de Processamento de Dados do Estado do Rio Grande do Sul, was mainly designed with the purpouse of supporting the demands for a low cost communication software that could link the various state departments amoung themselves. Because of its characteristics, the system is susceptible to potential security failures. Direto is based on free software, thus reducing project costs and enabling it to be used by any kind of organization with no cost, with independence of platform and upgrade flexibility.

The proposed model and developed prototype aim to guarantee the security of information transfered by the eletronic mail module of Direto, extending its interface and incorporating the concepts of electronic certification and public-key infrastructure. Thus implementing the five basic itens of security.

Keywords: Electronic certification, public-key infrastructure, Direto, information security.

1 Introdução

1.1 Motivação

A comunicação, parte essencial da vida do ser humano e que foi a base do seu progresso, tem crescido muito nos últimos tempos. A diversidade de tecnologias (correio eletrônico, telefone, rádio, televisão, Internet, e outras) tem contribuído para a incrível velocidade com que as informações são transmitidas. Hoje, tem-se notícias instantâneas sobre o que acontece em qualquer parte do mundo.

A Internet é um dos meios de comunicação que mais cresce em número de usuários, chegando a ser um fenômeno de massa, com milhões de usuários espalhados pelo mundo, movimentando milhões de dólares em comércio eletrônico. Há vários fatores que colaboram para que esse crescimento aconteça, um deles é o fato de sua tecnologia ser barata e aberta, tendo sido rapidamente incluída em todos os sistemas operacionais.

A Internet teve seu início no final dos anos sessenta [FER2003], quando quatro universidades americanas (a Universidade UCLA - em Los Angeles, o SRI - Stanford Research Institute, a UCSB - Universidade da Califórnia em Santa Barbara e a Universidade de Utah) apoiadas e financiadas pelo governo americano através da "Advanced Research Projects Agency" (ARPA) desenvolveram uma rede estratégica de computadores que pudesse sobreviver a uma guerra nuclear. Para atingir o objetivo principal, os especialistas definiram que os computadores não poderiam ficar próximos fisicamente, assim como deveriam estar disponíveis para conexão todo o tempo, já que dessa forma, qualquer computador da rede poderia acessar informações de outros, antes que uma catástrofe acontecesse. Nascia então a ARPANet com aproximadamente 500 minicomputadores. A Internet cresceu, até o início dos anos oitenta, na velocidade de um servidor a cada três semanas. Hoje a Internet tem mais de 100.000.000 de usuários, e continua crescendo vertiginosamente.

A Internet foi projetada, inicialmente, com o objetivo de permitir a conectividade entre as partes que estavam interagindo. Com esse propósito, concretiza um grande sonho da humanidade, a globalização, porém deixa grandes brechas em relação à segurança das informações trafegadas. Enquanto alguns usuários utilizam a Internet com a finalidade de se comunicar rapidamente, outros a utilizam para vasculhar as informações que estão trafegando. Essas pessoas são os *hackers* – pessoas estudiosas que se dedicam a experimentar os limites de um sistema por estudo, curiosidade ou até prazer, e os *crackers* – que são *hackers* que utilizam a sua vasta experiência para causar danos e obter vantagens.

O correio eletrônico é, sem dúvida, um dos mais importantes serviços utilizados na Internet. Quando usuários corporativos utilizam o correio eletrônico para trocar informações com outros usuários, eles podem estar expondo segredos da corporação para espionagem ou outros atos ilícitos realizados por *crackers.* Tipicamente, dados transitam sobre a Internet desprotegidos, atravessando múltiplos dispositivos e servidores de rede, sob vários *links* de comunicação insegura. Este dado é vulnerável a um agressor que queira

ver os dados transitando para extrair informações. Alternativamente, o agressor pode modificar, alterar ou excluir dados para confundir o receptor da mensagem ou pode personificar uma pessoa legítima para realizar funções que causem perdas para a entidade.

Com a crescente troca de informações sigilosas, como documentos válidos legalmente e transações de negócios, cresce cada vez mais a preocupação com a segurança no tráfego destes dados [WEB2000].

Para que haja confiança no momento da transmissão dos dados, o sistema deve garantir a autenticação: o receptor assegura quem é o responsável pelo envio dos dados, a integridade: garante que as informações recebidas são exatamente as mesmas que foram enviadas, a confidencialidade: assegura que apenas os destinatários poderão ter acesso às informações de forma legível quando estas chegarem ao seu destino, e o não-repúdio: que impede que as partes envolvidas neguem seu envolvimento no processo.

Para possibilitar que esses serviços sejam garantidos, técnicas de criptografia como a criptografia de chave simétrica e a criptografia de chave pública [BUR2002] foram criadas. A criptografia de chave simétrica é um método bem desenvolvido e eficiente, porém, tem um problema muito sério de distribuição de chaves. Criptografia de chave pública é menos eficiente na criptografia de grandes mensagens, porém resolve o problema da distribuição das chaves. A combinação dos dois métodos tem provido uma forma moderada de comunicação segura sobre a Internet.

A criptografia de chaves públicas, sozinha, não é suficiente para garantir a confiança de um sistema. Assim, com o propósito de garantir todos os serviços de segurança, surgiu a Infra-estrutura de Chaves Públicas (ICP). A regra principal de uma ICP é estabelecer a autenticidade de quem é confiável. Isto é feito com o uso de um certificado de chave pública na forma de um certificado digital, que associa uma chave pública a seu dono e é assinado por um terceiro confiável, chamado de Autoridade Certificadora (AC).

Com o aumento da preocupação com a segurança das informações, muitos desenvolvedores tem incorporado segurança a seus produtos, criando soluções próprias. Estas soluções acabam se tornando caras e difíceis de serem implementadas já que geram produtos não interoperáveis, sendo impossível integrá-los com produtos de outros desenvolvedores. Para que isto não ocorra é necessário o entendimento e o uso correto dos padrões e protocolos já publicados para a criação e gerenciamento do ciclo de vida de certificados digitais de forma totalmente interoperável com outros produtos disponíveis no mercado.

O software Direto, desenvolvido pela Procergs, [PRO2000] tem como objetivo principal atender a demanda de um software de comunicação de baixo custo, que interligue os diversos órgãos do estado. Por suas características o sistema está suscetível a potenciais falhas de segurança. O módulo de correio eletrônico funciona como um serviço de correio eletrônico normal, com as facilidades da Internet. Neste módulo o usuário pode ler, enviar, receber e encaminhar mensagens para qualquer contato, dentro ou fora da empresa. O tráfego de informações neste sistema pode variar desde mensagens sem nenhuma importância trocadas entre funcionários, até mensagens sigilosas trocadas entre os diretores

das várias organizações do Estado que já utilizam ou virão a utilizar o Direto como ferramenta padrão de correio eletrônico.

Visando garantir a segurança das informações trafegadas está sendo proposto um modelo que, estendendo a interface de correio eletrônico já existente, atenda os quatro quesitos básicos de segurança, incorporando os conceitos de certificados eletrônicos e infra-estrutura de chaves públicas.

1.2 Descrição dos Capítulos

Com o intuito de descrever o trabalho realizado, este texto está dividido da seguinte forma:

- **Capítulo 2:** realiza uma revisão dos conceitos básicos de uma Infra-estrutura de Chaves Públicas, descrevendo seus serviços, características, funcionalidades e componentes. É descrito, também, como os componentes se relacionam, realizando os processos necessários para a criação, assinatura e até o uso dos certificados;

- **Capítulo 3:** apresenta uma série de ferramentas e tecnologias estudadas, que foram utilizadas ou não para o desenvolvimento da infra-estrutura de chaves públicas para o Direto, procurando sempre destacar as razões que motivaram a escolha.

- **Capítulo 4:** apresenta uma descrição do projeto Direto, com uma breve descrição das tecnologias e ferramentas utilizadas para o seu desenvolvimento;

- **Capítulo 5:** apresenta uma proposta de solução para o problema de falta de segurança nas mensagens trafegadas pelo Direto. É apresentado todo o processo de criação e uso da infra-estrutura, desde a avaliação da necessidade da criação da ICP, passando pela criação da Autoridade Certificadora, finalizando com o uso do certificado pelos usuários do Direto.

- **Capítulo 6:** descreve o protótipo implementado, relacionando as principais características pretendidas para o mesmo. Apresenta o ambiente e as ferramentas utilizadas para o seu desenvolvimento. Finalmente, apresenta a estrutura de implementação e a interface disponibilizada junto ao Direto para a cifragem e assinatura das mensagens e a validação dos certificados;

- **Capítulo 7:** apresenta testes realizados para a validação da ferramenta, mostrando maneiras de comprometimento das mensagens e como a proposta apresentada consegue eliminar os riscos deste comprometimento;

- **Capítulo 8:** aponta novos caminhos e possibilidades que podem ser explorados a partir do modelo proposto. A descrição de trabalhos futuros serve como ponto de partida para o surgimento de novas propostas ou implementações;

- **Conclusão:** resume os problemas encontrados e as soluções alcançadas através do projeto e do desenvolvimento do protótipo da Infra-estrutura de Chaves Públicas da Procergs.

2 Infra-estrutura de chaves públicas

Neste capítulo será feita a descrição dos serviços, características, funcionalidades e componentes necessários para a criação de uma infra-estrutura de chaves públicas. Em seguida, será explicado como os componentes se relacionam, realizando os processos necessários para a criação, assinatura e até o uso dos certificados.

2.1 Definição

Infra-estrutura de chaves públicas, ou ICP, é uma arquitetura segura que combina software, hardware, procedimentos operacionais, regulamentação institucional (ou legislação), tecnologias de criptografia e serviços, e que tem sido introduzida para prover um aumento de confidencialidade na troca de informações sobre um ambiente inseguro [IBM1999].

O termo ICP é geralmente utilizado para descrever os mecanismos, as entidades, as políticas e os relacionamentos que são criados com o objetivo de recuperar e associar chaves públicas de criptografia aos seus donos.

2.2 Componentes

Uma ICP é criada para servir como uma estrutura básica de segurança de um sistema e é formada pela combinação de pessoas, serviços, padrões e tecnologias. Na figura 2.1 tem-se uma visão geral de uma ICP e de seus principais componentes, segundo [HOU1999].

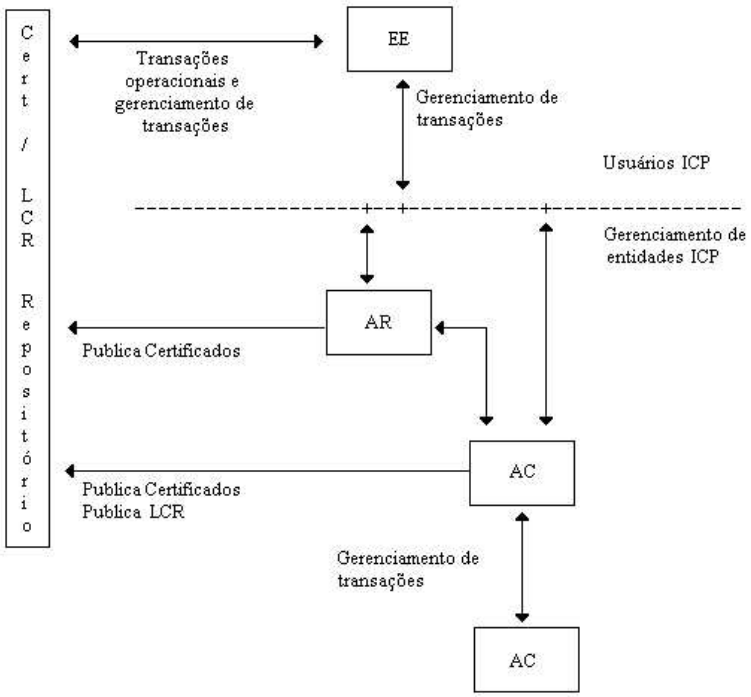

FIGURA 2.1 – Principais componentes de uma ICP

2.2.1 Entidades Finais

Uma entidade final (EF) é definida como um usuário da ICP. Entidade final é um termo genérico para um sujeito que utiliza alguns serviços ou funções de uma ICP, que pode ser o dono de um certificado (uma pessoa, uma organização ou alguma entidade), ou um requisitante (uma aplicação) para certificados ou listas de certificados revogados.

2.2.2 Autoridade Certificadora

A Autoridade Certificadora (AC) é a assinante do certificado. Ela tem a responsabilidade de identificar o sujeito do certificado. As operações básicas de uma AC incluem: criação de certificados, renovação de certificados, anulação de certificados, alteração dos estados dos certificados, e publicação de certificados e listas de certificados revogados.

A AC é uma entidade idônea, também conhecida como terceiro confiável, e é a responsável por garantir a autenticidade dos usuários, fazendo isso através da assinatura dos certificados com sua chave privada. Com isso, sua chave privada se torna um alvo natural para tentativas de comprometimento, já que, com ela, é possível criar certificados falsos com a intenção de se fazer passar por outra pessoa.

Uma AC é reconhecida por dois atributos: seu nome e sua chave pública.

2.2.3 Autoridade Registradora

Uma Autoridade Registradora (AR) é um componente opcional em uma ICP. A AR é uma entidade final confiável que é certificada pela AC, e que atua como um servidor subordinado da AC. A AC pode delegar algumas de suas funções de gerenciamento para a AR. Por exemplo, a AR pode realizar tarefas de autenticação pessoal, reportar certificados revogados, gerar chaves ou arquivar pares de chaves.

Uma AR pode aparecer para um requisitante de certificados como uma AC, mas ela não assina os certificados, é apenas um interface com a AC.

2.2.4 Repositório

Um repositório de certificados (RC) é o local onde são armazenados os certificados criados. A principal tarefa do repositório é prover dados que permitam aos usuários confirmar o estado de determinado certificado digital. Os tipos de repositórios comuns conhecido são descritos a seguir.

2.2.4.1 Diretórios

Diretórios são bancos de dados otimizados, especializados em leitura e pesquisa.

Um dos primeiros modelos de Diretório Público [FRE2003] foi criado em 1988 pela ISO. Este modelo de diretório resultou na recomendação X.500, e define um protocolo de acesso a diretórios e o método como os dados são armazenados e gerenciados. A recomendação X.500 não se tornou um padrão para a Internet, devido a incompatibilidade de seu modelo com os protocolos utilizados na Internet.
.

Devido a necessidade de acessos [OPE2003] a diretórios através da Internet foi projetado um novo serviço para ser utilizado em redes TCP/IP, baseado nos serviços de diretórios X.500. O protocolo foi denominado LDAP (*Lightweight Directory Access Protocol)*. Este protocolo organiza seus registros de forma hierárquica e seus acessos utilizam funções semelhantes as usadas em banco de dados. O LDAP tornou-se, em pouco tempo, o modelo mais utilizado para acesso de diretórios na Internet.

Um diretório pode ser público ou pode ter sua disponibilidade privada para uma finalidade específica. Um diretório é criado privado quando ele contém informações que não podem ser compartilhadas fora da organização

O protocolo LDAP permite alta flexibilidade no gerenciamento de certificados dentro de uma organização. Dados relevantes para o gerenciamento de certificados podem ser armazenados em diretórios LDAP. Por exemplo, uma AC pode utilizar informações de um diretório para popular um certificado com o nome da empresa ou outras informações. Dependendo da política de segurança da empresa, uma AC pode utilizar as informações do diretório de formas diferentes, criando certificados diferentes, de acordo com o seu uso.

2.2.4.2 E-mail e disquetes

Certificados podem ser enviados por e-mail ou copiados e entregues em disquete, para que o receptor possa adicioná-lo para o seu próprio conjunto de chaves em seu servidor ou desktop.

2.2.4.3 Bancos de Dados

Um banco de dados [ART2003] pode ser configurado para aceitar certificados no formato X.509. Este método pode ser utilizado por sistemas privados onde os métodos de pesquisa para localização de certificados não precisem seguir a estrutura LDAP.

A forma como um cliente se conecta e gerencia um banco de dados varia muito de acordo com o produto. Por esse motivo, o método de armazenamento de certificados em bancos de dados não é utilizado para infra-estruturas públicas, onde o acesso precisa ser padronizado.

2.2.5 Chaves

O termo chave refere-se a alguma informação (geralmente um número binário de tamanho especifico) que é usado em conjunto com um algoritmo de criptografia para as propostas de cifrar e/ou decifrar dados. Durante a fase de cifragem, a chave define o modo como os dados são embaralhados. Na fase de decifragem, apenas o uso da chave correta pode converter os dados cifrados nos dados de entrada correspondentes.

Na criptografia de chave pública, usada por uma ICP, são usados um par de chaves matematicamente relacionadas. Se uma chave é utilizada para cifrar uma informação, então apenas a chave relacionada pode decifrá-la. Se uma das chaves é conhecida, deve ser difícil calcular a outra chave relacionada. O par de chaves utilizadas pode ser descrito como:

- chave privada: é uma chave única e privada. Ela somente é conhecida pelo seu dono e não pode ser compartilhada com outros usuários, assim, a chave privada de um usuário permite que ele prove que é quem ele diz ser;

- chave pública: é a chave relacionada com a chave privada, porém ela é livremente distribuída e pode ser vista por todos os usuários.

Como uma regra geral, um par de chaves pode permanecer válido por um longo período de tempo, entretanto, existem algumas razões para limitar o tempo de vida do par de chaves:

- quanto mais longo o tempo de vida da chave usada, maior é a chance de ela ser comprometida;

- se a chave é comprometida, quanto maior é o tempo de vida da chave, maior as perdas que se tem. Se uma chave de criptografia é usada por um longo período de tempo, um grande número de documentos foram cifrados com ela e um grande número de documentos podem ser descobertos por um espião;

- avanços na tecnologia podem apresentar uma chave mais segura no futuro do que quando ela foi criada.

Dependendo da política de segurança em uso, um novo par de chaves pode requerer que todos os documentos cifrados previamente precisem ser decifrados e cifrados novamente com a nova chave. Qualquer documento assinado com a velha e expirada chave privada pode necessitar ser assinado com a nova chave.

Por essa razão, um par de chaves pode permanecer válido por um período de cinco anos ou mais, porém, alguns pares de chaves críticas podem ter seu tempo de vida reduzido para alguns meses, ou até menos. Uma política de segurança deve especificar o tempo de vida das chaves bem como as ações a serem tomadas quando os pares de chaves são alterados.

2.2.6 Assinatura Digital

Assinaturas digitais com chave pública provêem autenticação, integridade de dados, e não repúdio. Ela é superior a uma assinatura manuscrita pois é quase impossível falsificar.

A informação é embaralhada pelo usuário usando sua própria chave privada. Uma assinatura não pode ser forjada, supondo-se que só o dono conhece sua chave secreta. Com isso, o armazenamento da chave privada torna-se um dos pontos fundamentais para o correto funcionamento e a confiança em uma assinatura digital. Um bom mecanismo de armazenamento de chave deve ser adotado, podendo ser através de:

- proteção por senhas: forma mais comum empregada pelas ICPs. Uma senha ou PIN (personal identification number) é utilizado para cifrar a chave privada, que pode ser armazenada de forma segura. A segurança deste método consiste na utilização de uma boa senha, que não seja fácil de ser adivinhada;

- cartões PCMCIA (Personal Computer Memory Card International Association) : opção de armazenamento de chave em placas com um *chip*. Para a utilização da chave, ela deve ser transportada do cartão para a memória do sistema e, assim, continua sendo vulnerável à roubos;

- *tokens:* dispositivo de hardware que armazena a chave de forma cifrada;

- biometria: a chave é associada a alguma qualidade única que identifica um usuário (por exemplo, sua impressão digital);

- cartões inteligentes. A chave é armazenada em um cartão á prova de falsificação, que contém um *chip* de computador.

A assinatura digital é um processo lento e que aumenta o tamanho da informação original. A figura 2.2 mostra o processo realizado para a criação e verificação de uma assinatura digital.

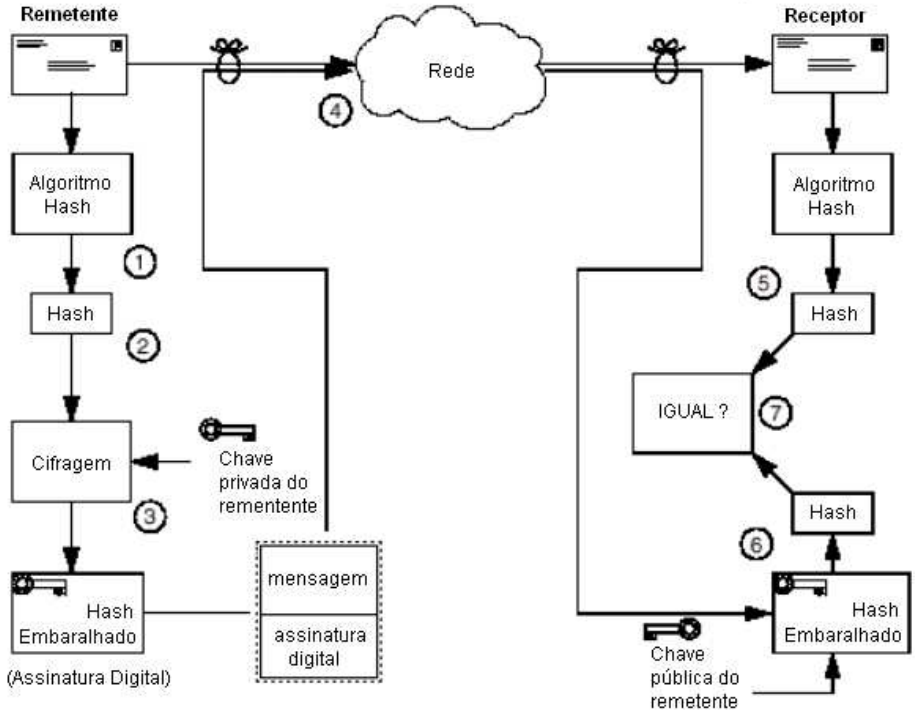

FIGURA 2.2 - Criação e verificação de uma assinatura digital

Os processo mostrado na figura 2.2 são descritos como:

1) o assinante aplica uma função *hash* na mensagem;

2) algoritmos *hash* podem gerar um *hash* de 128 ou 160 bits;

3) o valor *hash* é cifrado (o capítulo 3.1 Criptografia descreve os métodos e alguns algoritmos de criptografia) usando a chave privada do requisitante. O resultado é um valor *hash* cifrado, conhecido como assinatura digital;

4) o requisitante transmite a mensagem original juntamente com a assinatura digital para o receptor;

5) o receptor aplica a mesma função *hash* usada pelo remetente. O resultado é um valor *hash;*

6) o receptor usa a chave pública do remetente para decifrar o valor *hash* recebido junto com a mensagem;

7) o receptor realiza uma comparação entre os dois valores (o criado e o recebido) para determinar se a mensagem original foi alterada ou não e verifica se a assinatura é autêntica, isto é, o remetente é original.

Um sistema de assinatura digital [WEB2001] possui as seguintes propriedades:

- a assinatura é autêntica: quando o receptor utiliza a chave pública do remetente ele confirma que foi o remetente e somente o remetente quem assinou a mensagem;

- a assinatura não pode ser forjada: somente o remetente conhece sua chave privada, e ninguém mais pode assinar o documento no seu lugar;

- o documento assinado não pode ser alterado: se houver qualquer modificação na mensagem, a função *hash* aplicada pelo receptor gerará um resultado diferente da função *hash* aplicada pelo remetente;

- a assinatura não é reutilizável: a assinatura é uma função do documento e não pode ser transferida para outro documento;

- a assinatura não pode ser repudiada: o remetente não pode posteriormente negar ter assinado o documento.

.

2.2.7 Certificados

Certificados são estruturas de dados que atestam a associação de uma chave pública a um usuário final. A associação é afirmada por haver uma AC de confiança que assina digitalmente cada certificado. Um usuário confia em uma AC quando ela utiliza-se de processos e critérios para o gerenciamento de certificados bem definidos e públicos, mantém sua própria chave privada bem armazenada e não tem qualquer interesse comum com as partes envolvidas no processo.

Quando um remetente quer enviar uma mensagem para um destinatário, ele precisa anexar o seu certificado à mensagem. O receptor recebe a mensagem com o certificado e então verifica a assinatura da AC do certificado. Se a assinatura é de uma AC que o receptor confia, então ele pode confiar que a chave pública contida no certificado é realmente do assinante.

A recomendação X.509 do ITU (International Telecommunication Union), que foi publicado em 1988 como parte das recomendações de Diretórios X.500, define um formato padrão de certificados.

Dados em um certificado são escritos em ASN.1 (Abstract Syntax Notation) que depois são convertidos em dados binários no formato DER (Distinguished Encoding Rules). Esta operação permite que os dados dos certificados se tornem independentes para

cada plataforma. Detalhes sobre as notações ASN.1 e DER são descritas no capítulo 3.9: Representação de Dados.

Em alguns campos do certificado, um OID (Objeto Identifier) é usado para representar um valor específico. Um OID é projetado para tratar da possibilidade de reutilização desses campos com o passar dos anos. A figura 2.3 mostra a estrutura básica de um certificado X.509.

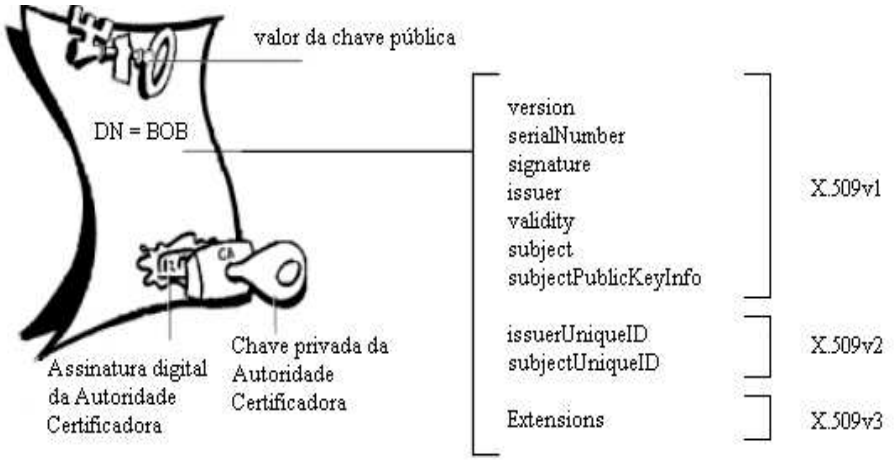

FIGURA 2.3 - Certificado

Extensions são novos campos que foram introduzidos a partir da versão X.509v3 e adicionam flexibilidade para o certificado. Cada campo *extension* contém um campo do tipo *boolean* que define se ele é um campo crítico ou não. Se um campo é marcado como crítico, quem o recebe precisa reconhecer este campo, ou então, deve rejeitar o certificado.

2.2.8 Lista de Certificados Revogados

Uma Lista de Certificados Revogados (LCR) é uma lista datada e assinada pela AC e tornada livremente disponível em um repositório público, como um meio de notificar clientes que precisam verificar a revogação de um certificado. As circunstâncias que podem tornar nulo um certificado incluem:

- o dono de um certificado é removido do domínio de segurança;
- o dono de um certificado reconhece que sua chave privada foi perdida ou comprometida;
- o usuário esquece a senha para a sua chave privada (quando o armazenamento utilizado faz uso desta característica);
- alguma informação contida no certificado se torna incorreta;

- surgem suspeitas de ataque de segurança a um certificado ou chave privada.

Existem formas diferentes de se manter e publicar uma LCR. A implementação recomendada, entretanto, é armazenar a LCR em um diretório LDAP. O padrão PKIX (conjunto de padrões e protocolos definidos para garantir o suporte necessário ao desenvolvimento de uma infra-estrutura baseada em certificados X.509) permite que o ponto de distribuição da LCR seja colocado em uma extensão do certificado chamado de *CRL Distribuition Points*. As aplicações podem então buscar esta informação de localização (por exemplo, a URL) do certificado para acessar a LCR.

2.3 Serviços

Uma ICP deve garantir os quatro serviços básicos de segurança, que são:

- autenticação: é o processo que garante que uma pessoa é realmente quem ela diz ser;

- confidencialidade: determina que apenas as pessoas autorizadas podem acessar as informações. Confidencialidade de dados também é referenciada como privacidade ou sigilo;

- integridade: é a garantia de que os dados não foram alterados ou destruídos de maneira não autorizada;

- não repúdio: é a garantia que o remetente não pode negar que é o criador da mensagem, e que o receptor não pode negar que recebeu a mensagem;

- datação: permite determinar se um documento eletrônico foi criado ou assinado em uma determinada data/hora

.

2.4 Características e Funcionalidades

Algumas características e funcionalidades [ARC1999] também são esperadas de um ICP:

- gerenciamento do ciclo de vida da chave: o ciclo de vida de uma chave depende de como ela será usada, se fornecendo confidencialidade ou não-repúdio. O gerenciamento deste ciclo de vida deve levar em conta:

  - os métodos de geração de chaves devem ser prudentes, sujeitos a decisões comerciais e requerimentos de negócios. Para a seleção da chave deve-se levar em conta a qualidade, a unicidade, a discrição e a recuperabilidade.

- gerenciamento do ciclo de vida dos certificados: uma ICP deve fornecer as seguintes funcionalidades:

  - distribuição das chaves para esquemas e diretórios de armazenamento apropriados

  - habilidade de uma AC anular certificados

  - habilidade de uma AC para suspender e reativar certificados, desde que observados os termos da política aplicada

  - habilidades da AC de forçar a entrega de notificações sobre a anulação, suspensão e reativação das chaves

  - permitir ao usuário repudiar sua chave pública, desde que observados os termos da política aplicada

  - permitir ao usuário e assinante recuperar notificações sobre a anulação, suspensão e reativação de chaves

  - permitir ao usuário ou assinante determinar o estado de um específico certificado (revogado, suspenso, ...)

  - permitir o arquivamento e subseqüente recuperação de certificados como suporte à recuperação e verificação de informações, desde que observados os termos da política aplicada. A recuperação de certificados deve fornecer implementações que permitam diferentes cenários, quer seja recuperação por força da lei, recuperação por parte da corporação e recuperação por parte de indivíduos (que são sujeitas à política aplicada e autorizações);

- segurança da ICP: uma ICP, por si só, também deve ser segura, tendo em conta os seguintes requisitos:

  - proteção da confidencialidade, integridade e disponibilidade dos serviços da própria ICP

  - prover serviços de não-repúdio para ações de serviços de certificações

  - prevenir-se contra usuários e assinantes sobre o repúdio de suas ações

- interoperabilidade: os elementos distribuídos por diferentes vendedores devem fornecer interoperabilidade. Os elementos devem:

  - fornecer padrões internacionais para certificados e dados associados

  - fornecer padrões internacionais para serviços de certificação

- fornecer internacionalização de todos os certificados e dados associados

- fornecer internacionalização de todos os serviços de certificação.

2.5 Arquitetura

Donos de certificados podem obter seus certificados de diferentes ACs [PER1999], dependendo de como é a organização ou comunidade das quais são membros. A ICP é composta de muitas ACs ligadas por caminhos confiáveis

Uma das arquiteturas implementadas é a arquitetura física, que pode ser vista na Figura 2.4. Nesta arquitetura os maiores componentes da ICP são implementados em sistemas diferentes: a AC em um sistema, as ARs em sistemas diferentes e os servidores de diretórios em outros sistemas.

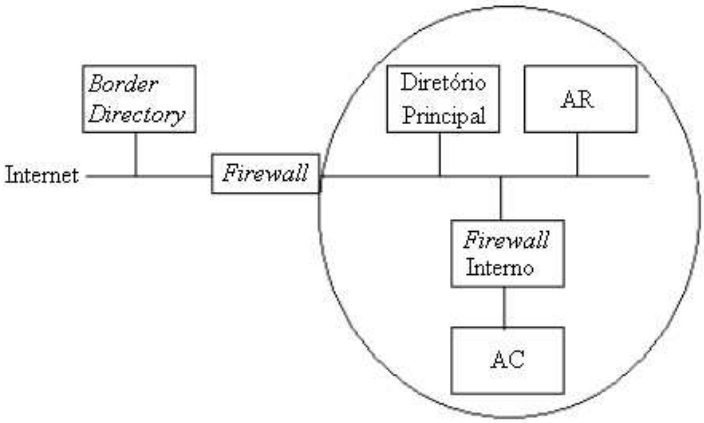

FIGURA 2.4 – Arquitetura física de uma ICP

Nesta arquitetura, os diretórios necessitam estar publicamente disponíveis, para que outras organizações possam acessar os certificados. Como o servidor de diretórios pode ser utilizado para outras aplicações, contendo dados sigilosos, uma solução a ser adotada pode ser criar um diretório que contenha apenas as chaves públicas ou certificados e localizá-lo na fronteira da organização – este diretório é referenciado como *border directory*. Uma provável localização para o diretório pode ser fora do *firewall* da organização ou periferia, em um segmento DMZ (segmento desmilitarizado) protegido na rede. O segmento DMZ não pode ter acesso a rede corporativa da empresa, como garantia contra possíveis ataques, sendo permitido somente acessos a serviços que estejam habilitados e definindo regras para estes acessos, para que ele esteja disponível para o público, mas mantendo-se protegido.

O sistema da AC é particularmente importante porque um comprometimento neste sistema pode comprometer todas as operações da ICP. Por isso, proteger o sistema da AC com um *firewall* adicional é recomendado, de tal forma que ele fique protegido contra outros sistemas, dentro da própria organização.

O *border directory* deve ser periodicamente atualizado com os novos certificados criados ou com as atualizações de certificados realizadas no diretório principal. Os usuários da organização irão usar o servidor de diretório principal, enquanto que outros sistemas ou outras organizações só terão acesso ao *border directory*.

2.6 Ciclo de Vida de um certificado

Uma organização necessita de uma política de segurança para determinar o ciclo de vida de um certificado. Este ciclo pode, também, ser gerenciado por qualquer acordo que o usuário assine com a AC. Por exemplo, se o usuário provê informações falsas a AC pode ter que anular o certificado do usuário. Na figura 2.5 é mostrado o possível ciclo de vida de um certificado. As linhas sólidas indicam o ciclo de vida normal de um certificado. As linhas tracejadas indicam quando uma AC ou AR precisam intervir no ciclo de vida.

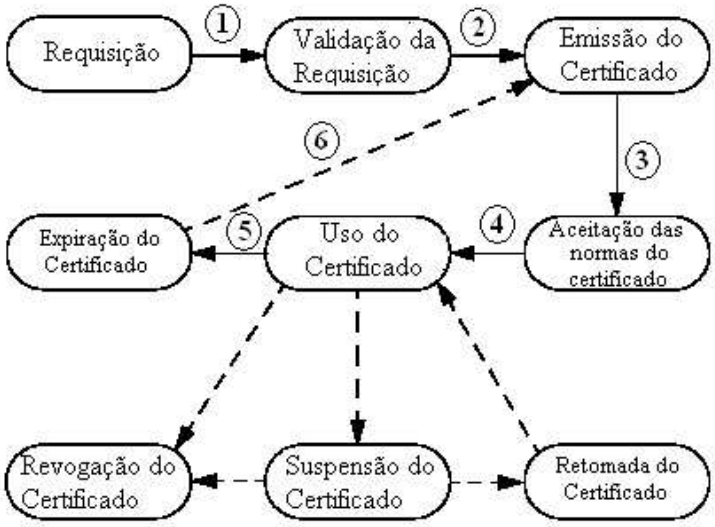

FIGURA 2.5 – Ciclo de vida esperado de um certificado

Os fundamentos do ciclo de vida de um certificado serão detalhados à seguir.

2.6.1 Emissão

Um componente AC é o responsável pela criação e distribuição dos certificados.

Um usuário deve ser positivamente identificado antes que seja permitido gerar e distribuir um certificado para ele. Dependendo da implementação, um usuário pode ser requisitado a preencher um formulário on-line na aplicação, onde questões básicas como nome e endereço pessoal podem ser perguntados, de forma a estabelecer a identidade do usuário. O nível da informação requisitada pela aplicação pode depender do tipo de certificado que está sendo criado, sua aplicação e da política de segurança adotada. Autenticação para um certificado de e-mail, por exemplo, dependendo da política aplicada, pode requer apenas que o usuário possa receber e-mail no endereço por ele fornecido, assim, o usuário seria requisitado apenas a fornecer um endereço válido.

Uma vez que detalhes do usuário devem ser verificados por um componente AR (se ele for implementado), uma requisição deve ser enviada da AR para a AC para a emissão do certificado. A AC pode receber a requisição na forma de *template*, que já inclui a chave pública a ser certificada. A AC pode então aplicar sua chave de assinatura apropriada, efetivamente assinando a chave pública. Os campos relevantes do *template* podem ser alterados pela AC e o certificado é enviado para a AR. A AC pode também guardar uma cópia local do certificado. Ao invés do certificado ser entregue ao usuário, ele pode ser publicado para um diretório público.

Uma AC que assina e emite um certificado não necessariamente torna este certificado válido. Algumas ACs requerem que o usuário do certificado primeiro aceite certas normas de acordo com sua política de segurança, antes que seu certificado seja publicado. Estas ACs criam “certificados operacionais” para os usuários que requisitam um certificado. Na prática, os certificados podem ser enviados para o usuário com a norma a ser aceita, requisitando ao usuário a não usar o certificado até que o acordo seja formalmente aceito. Assim que a AC recebe a aceitação do usuário, ela publica o certificado em um repositório público, tornando-o válido.

2.6.2 Suspensão

Em algumas situações específicas e quando determinado pela política de segurança de uma organização, um certificado pode ser colocado em estado de espera, ou ser suspenso. Um exemplo dessa situação pode ser o período de férias de um funcionário da organização, que quer ter a certeza que seu certificado não será utilizado para operações fraudulentas. Na volta das férias, seu certificado é retomado, voltando a ficar em um estado válido para uso, ou é anulado necessitando ser recriado.

2.6.3 Revogação

Quando um certificado é criado, a expectativa é que ele seja usado por todo o seu período de validade. Entretanto, várias circunstâncias podem tornar um certificado inválido antes do período de expiração. As circunstâncias podem ser: troca de nome, troca da associação entre o sujeito e a AC (ex.: um funcionário que termina o contrato com a empresa), e o comprometimento ou suspeita de comprometimento da correspondente chave privada. Sobre estas circunstâncias, a AC precisa revogar o certificado.

Uma organização que deseja desenvolver uma ICP precisa determinar de quem ela pode aceitar um pedido de revogação de um certificado. O dono de um certificado pode requisitar sua revogação, por exemplo, se a chave privada do usuário for comprometida. Em outro caso, uma outra pessoa pode fazer essa solicitação. Por exemplo, um funcionário pode pedir ao seu gerente para revogar seu certificado. Uma outra forma é a própria AC pedir a revogação, se ela descobrir, por exemplo, que o dono do certificado infringiu a política de segurança da AC. Em situações onde a pessoa que requer a revogação não é o dono do certificado, deve haver uma forma de determinar se o requisitante tem a autoridade

necessária para requerer esta solicitação. Isto pode ser determinado pela política de segurança da empresa.

O padrão PKIX define um método de revogação de certificados. Este método estipula que cada AC deve, periodicamente, criar uma LCR.

Cada certificado revogado é identificado na LCR pelo número de série. Quando uma aplicação necessita utilizar um certificado, ela não verifica apenas a assinatura e a validade do certificado, mas também adquire a LCR mais recente e verifica se o número de série está na LCR. Uma AC cria uma LCR em períodos regulares.

Uma vantagem deste método de anulação é que a LCR pode ser distribuída da mesma forma que os próprios certificados, através de uma comunicação insegura. Uma limitação deste método é que a granularidade da revogação é limitada pelo período de criação da LCR. Se um certificado for revogado neste exato momento, a revogação só estará disponível no próximo período de criação da LCR – que pode ser uma hora, um dia, ou uma semana, dependendo da freqüência com que a AC cria as LCRs.

Um outro método de notificação de revogação pode ser o método de verificação on-line. Verificação on-line pode significar reduzir o intervalo entre a revogação e a distribuição da informação para as partes envolvidas. Assim que a AC aceita a requisição de revogação como válida e autêntica, qualquer consulta para o serviço on-line irá refletir corretamente os impactos daquela revogação. Neste método, dependendo da localização dos certificados e do modelo de confiança adotado, múltiplas consultas podem ser necessárias para verificar todos os certificados das ACs envolvidas no processo.

Um certificado revogado não pode mais ser utilizado por um usuário. O processo de emissão deverá ser iniciado, dando início ao ciclo de vida de um novo certificado..

2.6.4 Expiração

Cada certificado que uma AC entrega tem uma data de expiração. Uma AC pode entregar certificados com um padrão de período de validade. Entretanto, o período de validade pode depender das circunstâncias em que o certificado vai ser usado. Após a data de expiração, o certificado não deve mais ser usado para qualquer tipo de autenticação e/ou validação. Um usuário pode ser avisado pela AR (se implementada) da próxima data de expiração do certificado. Então, o usuário pode ser requisitado a seguir o processo de renovação.

O tempo de vida do certificado necessita ser descrito na política de segurança, que deve detalhar o tipo de classe de certificado que está em uso e sobre quais condições a AC (ou AR) deve interferir no seu ciclo de vida normal.

2.6.5 Renovação

Quando um certificado expira, o processo de renovação pode ser iniciado. Se o processo é bem sucedido, ele deve resultar no usuário recebendo um novo certificado, que tem uma nova data de expiração.

2.7 Documentos de Política

A política que uma organização define e implementa [BUR2002] pode determinar como uma ICP opera dentro da organização. A política atua como um pré-requisito para muitas atividades do ICP e detalhes de implementação. Políticas também definem como a ICP interage com o mundo externo. Por isso, torna-se muito importante verificar alguns dos documentos de política disponíveis para uma ICP e quais os fatores a considerar em sua construção.

Basicamente, existem dois tipos de documentos:

- uma política de certificados (PC): a política de certificados da AC determina qual certificado pode ser usado;

- uma Declaração de Práticas de Certificação (DPC): é a descrição de como a política de certificados pode ser implementada para uma organização específica.

Os detalhes de cada documento serão descritos a seguir.

2.7.1 Política de certificados (PC)

Uma política de certificados [CHO1999] é definido como “um determinado conjunto de regras que indica a aplicabilidade de um certificado para uma comunidade e/ou classe particular de aplicação com requerimentos de segurança comercial”. Uma política de certificado pode ser usada por um usuário que precisa decidir quando confiar em um certificado para uma proposta específica ou não. Por exemplo, uma política de certificado de uma organização pode requerer que o certificado criado para o usuário possa apenas ser usado para autenticação na *Intranet* da organização. A política de certificado deve ser reconhecida pelo usuário e pelo criador (AC). Modelos padrões tem estimulado o uso de OIDs nos campos de políticas nos *templates* dos certificados. Quem registra o OID também publica uma especificação textual da política de certificados para ser examinada pelos usuários de certificados.

Certificados digitais são usados como uma base para obtenção de uma comunicação confiável e segura entre as partes do processo. Por isso, uma política de segurança necessita descrever todas os possíveis relacionamentos criados para aquele ambiente. Uma política obscura ou inadequada que leva a um erro de implementação no sistema de segurança pode afetar a integridade de toda a ICP. As áreas básicas a serem consideradas numa política são:

- se uma AR fizer parte da operação de uma ICP, políticas adicionais devem ser definidas especificamente para este componente;

- criação de uma ou mais políticas de certificados e uma Declaração de Práticas de Certificação (DPC). Estas políticas tem que ser ajustadas com qualquer outra política de segurança da organização;

- as formas pela qual uma chave pode ser protegida;

- processo de como uma chave de AC comprometida pode ser manipulada;

- as formas como os repositórios da ICP podem ser protegidos;

- processo de como usuários e ACs devem ser verificados;

- alcance de operação da ICP;

- processo de como uma ICP pode ser auditada;

- processo de como as políticas da ICP podem ser aprovadas, implementadas e mantidas.

O desenvolvimento de uma ICP ampla pode envolver múltiplas ACs e, assim, de múltiplos domínios de segurança. De forma a tornar a ICP da organização inter-operável, os relacionamentos entre as ACs tem que ser definidos. Estes relacionamentos entre ACs criam um ou mais caminhos de certificação (*certificate path*). Uma AC pode também criar tipos diferentes de certificados para usuários, criando diversos relacionamentos de confiança. A organização deve decidir então, qual a rota de certificação ela deve seguir. Em uma situação simplória, uma organização cria certificados para seus próprios empregados. Todos os certificados manipulados confiam na mesma AC, então o caminho de certificação necessita apenas de um único certificado. A organização pode ter que decidir se deve apenas confiar em certificados de usuários em seu próprio domínio de confiança ou de outros domínios também, ou se vai confiar apenas em um conjunto limitado de domínios.

2.7.2 Declaração de Práticas de Certificação (DPC)

A RFC2527 define uma DPC como uma declaração de práticas que uma AC emprega para a criação de certificados [CHO1999]. Uma DPC pode ter a forma de uma declaração, pela AC, dos detalhes do sistema de confiança e das práticas que ela emprega em suas operações, e no suporte à criação de um certificado. Pode também ser parte de um contrato entre a AC e o assinante, ou pode também ser composta de múltiplos documentos (uma combinação de leis públicas, contratos privados e /ou declarações).

Os conceitos de política de certificados e DPC vêm de diferentes fontes e foram desenvolvidos por razões diferentes. Entretanto, sua inter-relação é importante. Uma DPC é uma declaração detalhada de uma AC e suas práticas, e que necessitam ser entendidas e

consultadas por assinantes e usuários de certificados. Apesar do nível de detalhes variar em uma DPC, ela geralmente é mais pormenorizada que uma política de certificados. Uma DPC pode ser totalmente compreensível, gerando documentos robustos, e descrições precisas e detalhadas dos procedimentos do gerenciamento do ciclo de vida da chave.

Uma DPC é uma forma da própria organização se proteger e posicionar seus relacionamentos de negócios para com assinantes e outras entidades.

A RFC2527 [CHO1999] define um conjunto de cláusulas padrão para uma DPC. Este conjunto básico de cláusulas pode ser definido como:

- Introdução: identifica e introduz o conjunto de cláusulas, e indica os tipos de entidades e aplicações para a qual a especificação é destinada;

- Disposições gerais: especifica um conjunto de intenções de tópicos gerais e legais;

- Identificação e Autenticação: descreve os procedimentos adotados para autenticar um candidato, aplicado por uma AC ou RA antes de criar o certificado. Também descreve como os requisitantes de renovação e anulação são autenticados;

- Requerimentos operacionais: este componente é usado para especificar requerimentos impostos sobre a AC emissora, assinantes de ACs, ARs ou entidades finais, com respeito às várias atividades operacionais;

- Controles de segurança física, de procedimentos e pessoais: este componente descreve controles de segurança não técnicos usados pela AC emissora para realizar seguramente as funções de geração de chaves, autenticação de assinantes, emissão de certificados, anulação de certificados, auditoria e arquivamento;

- Técnicas de controle de segurança: este componente é usado para definir as medidas técnicas de segurança tomadas pela AC emissora para proteger as chaves de criptografia. Este componente pode impor obrigações em repositórios, assinantes de ACs e entidades finais para proteger suas chaves de criptografia e os parâmetros críticos da segurança. Este componente também descreve outros controles de segurança técnicos usados pela AC emissora para realizar seguramente as funções de geração de chaves, autenticação de usuário, registro de certificados, anulação de certificados, auditoria e arquivamento;

- Perfil de certificados e LCRs: este componente é usado para especificar formatos de certificados e, se uma LCR é utilizada, o formato da LCR. Assumindo o uso do certificado X.509, esta especificação inclui informações de modelos, versões e extensões usadas;

- Especificações de administração: este componente é usado para especificar como a política de um certificado em particular ou a definição da DPC pode ser mantida.

2.8 Administração de uma ICP

A administração de uma ICP descreve quais as atividades necessárias no desenvolvimento e na operação de uma ICP.

Pares de chaves e certificados são gerados e usados em diferentes épocas e armazenadas de formas diferentes sobre uma rede. Múltiplos pares de chaves podem ser gerados para diferentes usuários, sendo um par de chaves para propostas de criptografia, enquanto outro é criado para propostas de assinatura. Pares de chaves podem também ser criados dependendo da aplicação em que serão usadas e dos níveis de confiança associados com a aplicação. No caso da AC gerar os pares de chaves, uma forma de distribuição segura destas chaves deve ser definida.

Administração de uma ICP também envolve todos os processos de cópias de segurança e restauração das mesmas. Uma organização que distribui pares de chaves para seus funcionários pode necessitar armazenar uma chave em uma área segura de forma a decifrar certas informações quando necessário. Em uma situação extrema, se a chave da AC é perdida acidentalmente e a AC não possuir uma cópia, ela estará impedida de assinar qualquer certificado e LCR, e todo o domínio de confiança é perdido. O formato como as cópias são feitas deve ser definida (cifradas, em texto puro ou em partes de chaves). Os processos de cópia e restauração devem cobrir: geração de logs de auditoria, arquivamento de material, cópia e recuperação das chave privada da AC, AR e da entidade final.

A administração de alterações em chaves para uma ICP é outra tarefa importante. Uma organização pode decidir quando gerar e distribuir novos pares de chaves para os usuários. Uma alteração pode ser projetada para trabalhar semi-automatizada, ou ela pode ser projetada para que um usuário inicie o processo. Algumas chaves podem ser usadas para transações sigilosas e podem necessitar de alterações mais freqüentes que outras. A política de segurança pode decidir como a velha chave será destruída. Mesmo que a chave tenha expirado ela ainda pode ser usada para decifrar informações.

O gerenciamento de certificados inclui geração de certificados, revogação de certificados, administração da CRL e a criação e implementação de um modelo de confiança para a organização.

2.9 Considerações Finais

Este capítulo teve o propósito de revisar alguns conceitos básicos de uma Infra-estrutura de chaves públicas, procurando apresentar suas características e ressaltando a sua importância na obtenção de um aumento de confidencialidade na troca de informações sobre a Internet.

A descrição dos componentes e serviços da ICP teve o objetivo de mostrar a abrangência e o tamanho do trabalho envolvido na sua criação.

3 Ferramentas, Tecnologias, Representação de Dados e Padrões

Este capítulo detalha as ferramentas, as tecnologias, as diversas formas de representação de dados e os padrões estudados e utilizados, ou não, para o desenvolvimento da infra-estrutura de chaves públicas para o Direto, procurando sempre destacar as razões que motivaram a escolha.

3.1 Criptografia

Criptografia é a ciência de como cifrar e decifrar dados. Um dos principais objetivos da criptografia é garantir o armazenamento e circulação segura de mensagens.

Um algoritmo criptográfico é uma função matemática usada no processo de cifragem e decifragem. Um algoritmo criptográfico trabalha em combinação com uma chave – uma palavra, número ou frase – para cifrar o texto puro. O mesmo texto puro cifrado com diferentes chaves gera diferentes textos cifrados.

Neste capítulo serão analisados brevemente três categorias de algoritmos de criptografia: a criptografia convencional, a criptografia de chave pública e as funções de *hash* seguro.

3.1.1 Criptografia Convencional

Neste método de criptografia, existe uma única chave compartilhada entre o remetente e o receptor. A chave é uma peça secreta da informação que é usada para cifrar e decifrar a mensagem. Se a chave é secreta, então nenhum outro remetente ou receptor pode ler a mensagem.

Alguns algoritmos bastante utilizados para criptografia convencional são:

- DES: ciframento composto [CAR2000] que processa blocos de texto puro de 64 bits, produzindo blocos cifrados de 64 bits;

- IDEA: este ciframento [CAR2000] embaralha os dados em grupos de 64 bits e usa uma chave de 128 bits, que é o suficiente para resistir à maior parte dos ataques, o que o torna confiável e seguro.

- AES: O algoritmo AES é [DAE1999] o sucessor do algoritmo DES. Ele opera em quatro camadas: substituição de bytes, deslocamento de linhas, mistura de colunas e adição da chave. A substituição de bytes é um processamento não linear através de caixas S-Box (caixas de substituição não lineares que visam emaranhar o texto cifrado para que se torne mais difícil a sua decifragem). O deslocamento de linhas tem por objetivo fazer uma transposição dos blocos resultantes das caixas S-Box. A

mistura de colunas é obtida através de uma fórmula matemática com o objetivo de unir diferentes colunas de forma ordenada. A adição de chaves varia conforme o tamanho indicado na criptografia, após esta execução o processamento está pronto para fechamento da criptografia e geração do arquivo de saída.

3.1.2 Criptografia de Chave Pública

Neste método de criptografia, é criado um par de chaves: uma chave pública, que cifra os dados, e uma correspondente chave privada (chave secreta) que decifra os dados, ou vice-versa. A chave pública é publicada para o mundo enquanto a chave privada é guardada em segredo. Qualquer um com uma cópia da chave pública pode cifrar informações que apenas o dono da chave privada pode decifrar.

Alguns algoritmos conhecidos [CAR2000] para criptografia de chave pública são:

- RSA: este algoritmo pode ser utilizado para criptografia de informações e assinaturas digitais. As chaves podem ter vários tamanhos dependendo do tipo de implementação. É o mais popular algoritmo de chave pública, mais fácil de compreender e de implementar. Sua segurança está na dificuldade de fatorar grandes números: as chaves são calculadas matematicamente multiplicando dois números primos de grande tamanho. A grande desvantagem deste algoritmo é a lentidão do processo, que envolve a exponenciação modular (uma série de multiplicações);

- ElGamal: algoritmo baseado na exponenciação e na aritmética modular. ElGamal é usado preferencialmente para criptografia e assinaturas digitais.

3.1.3 *Hash* Criptográfico

Uma função *hash* pega uma entrada de tamanho variável e uma mensagem de qualquer tamanho e produz um produto de tamanho fixo, chamado de *message digest*. Não existe nenhuma maneira de retornar ao valor original com uma função *hash*, por isso, esse algoritmo é também conhecido como *one-way function* – função de mão única.

A função *hash* assegura que, se a informação é mudada de qualquer forma – até mesmo por um só bit – uma saída inteiramente diferente é produzida. O uso de um valor mínimo de *hash* de 128 bits é essencial para a segurança e um valor de 160 bits é ainda melhor.

Alguns algoritmos de *hash* seguro são:

- MD4: este algoritmo produz um *hash* de 128 bits, trabalhando sobre blocos de 512 bits, divididos em 16 sub-blocos de 32 bits. A saída do método é um conjunto de 4 blocos de 32 bits, que são concatenados para formar o *hash* de 128 bits. Como um todo, MD4 não pode ser atacado a não ser pelo método da força bruta (tentando-se

todas as combinações possíveis para a chave), mas diversos estágios internos já foram cripto-analisados individualmente com sucesso;

- MD5: é uma melhoria do MD4, com o acréscimo de mais funções de "embaralhamento" e mais rodadas;

- SHA: este algoritmo foi desenvolvido pelo governo dos Estados Unidos, em 1994. Ele foi inspirado no MD5, mas com a diferença de produzir um *hash* de 160 bits, em vez de 128 bits.

- SHA1: SHA-1 é uma revisão de SHA. SHA1 difere do padrão original SHA, que foi substituído, apenas na inclusão de 1 bit de rotação em seu método.

3.2LDAP

Em 1988, o CCITT, hoje conhecido como ITU-T, criou um padrão para serviços de diretório, o padrão X.500, que define um diretório que pode ser usado universalmente para grandes quantidades de dados.

Para acessar um diretório X.500, um cliente usa o *Directory Access Protocol* (DAP), definido junto com o padrão X.500, e que é um protocolo muito complexo e robusto, não podendo ser facilmente implementado em computadores de porte menor, como os *desktops*. X.500 foi então limitado a computadores grandes e para implementações de grande escala.

A Universidade de Michigan e a Netscape Communications Corp começaram, então, a criar uma simplificada versão do DAP, que foi chamada *de Lightweight Directory Access Protocol* (LDAP). LDAP implementa muitas das características do DAP, com a perda de algumas funções complexas e raramente usadas. Sua implementação é relativamente simples e ele pode, então, ser usado por aplicações desktop. LDAP é um protocolo que roda sobre TCP/IP. O protocolo padrão LDAP não apenas inclui definições de protocolo de baixo nível de rede, mas também representação de dados e funcionalidades de manipulação. O padrão LDAP não tem como meta definir como os dados são atualmente armazenados no diretório. A Figura 3.1 mostra as correspondentes traduções realizadas desde a requisição do cliente, até a consulta aos dados do diretório.

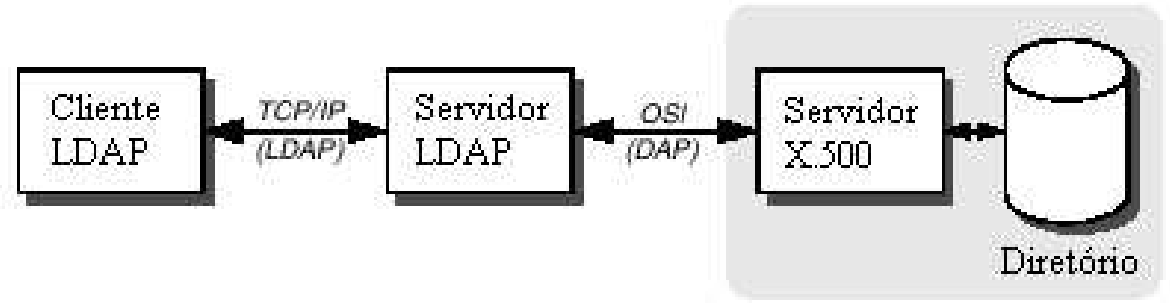

FIGURA 3.1 – Acesso ao diretório

Serviços de diretórios são úteis para uma grande gama de aplicações. Diretórios podem conter informações sobre pessoas, processos, recursos e grupos. Disponibilizando tais informações em uma área de armazenamento comum e através de protocolos de acesso abertos, diversas aplicações de instituições diferentes podem ter acesso a um conjunto de dados consistente.

O software Direto já utiliza o serviço de diretórios LDAP, implementado através de OpenLDAP, para armazenar os dados dos usuários do correio eletrônico. Esta implementação vai ser expandida para armazenar também a chave pública e o certificado dos usuários.

3.2.1 Modelos LDAP

O LDAP define 4 modelos básicos que descrevem por completo a sua operação, que informações podem ser armazenadas em diretórios LDAP e o que pode ser feito com essas informações. São eles:

- modelo de informação: define o tipo de informação que pode ser armazenada em um diretório LDAP. As informações são armazenadas em entradas. Uma entrada é uma coleção de atributos que contém um único DN (Distinguished Name). Cada um dos atributos das entradas tem um *type* e um ou mais *values*. Atributos podem ser valores-simples (por exemplo, uma pessoa tem apenas uma data de nascimento) ou multi-valorados (uma pessoa pode ter muitos telefones de contato). Os *types* são tipicamente *strings* mnemônicas (fáceis de lembrar), como “cn” para common name, ou mail para endereço de e-mail. A sintaxe dos valores depende do tipo de atributo. Por exemplo, o atributo “cn” pode ter o valor João Silva e o atributo e-mail pode conter o valor joao.silva@exemplo.com.br;

- modelo de nomes: define como a informação no diretório LDAP pode ser organizada e referenciada;

- modelo funcional: define o que pode ser feito com a informação no diretório LDAP e como ela pode ser acessada e alterada;

- modelo de segurança: define como a informação no diretório LDAP pode ser protegida de acessos ou modificações não autorizadas. Este modelo provê autenticação e identificação dos usuários que acessam os serviços do diretório. Isso é feito através da operação *bind*. Uma vez que o usuário é autenticado e identificado, as informações sobre controle de acesso podem ser consultadas para determinar se o usuário tem permissão para fazer o que ele solicitou. Há a possibilidade do usuário não se identificar, ou seja, acessar o diretório como anônimo. Nesse caso as regras de controle de acesso também determinarão o que o usuário poderá acessar no diretório.

3.2.2 Características do LDAP

Algumas características do LDAP são definidas como:

- escalabilidade: diretórios LDAP são altamente escalonáveis. Grandes diretórios são possíveis e com uma performance excelente;

- disponibilidade: LDAP fornece replicação e divisão de *namespace*. Replicação permite que múltiplos servidores LDAP possam armazenar o mesmo conteúdo, que podem ser disponibilizados quando um deles falha. Divisão permite que partes do diretório (*namespace*) sejam armazenados em servidores diferentes, aumentando a disponibilidade;

- segurança: características de segurança são oferecidas de forma que acessos não autorizados possam ser prevenidos. Protocolos de comunicação segura, como SSL, e mecanismos de autenticação garantem um alto nível de segurança. Isto torna possível o armazenamento de dados sigilosos em diretórios LDAP;

- maneabilidade: as novas versões do LDAP já possuem a facilidade da interface gráfica para facilitar a alteração do usuário, tanto para a parte de administração do sistema, como para a parte de administração do diretório de dados. Os usuários tem acesso aos serviços do LDAP através de interface gráfica, Web, linha de comando ou por APIs;

- padronização: o protocolo LDAP e muitas das funcionalidades da interface cliente/servidor, APIs e definições de dados são definidos como padrões oficiais ou correspondentes RFCs.

3.2.3 Implantando um serviço de diretórios distribuídos

Para implantar um serviço de diretórios distribuídos, algumas etapas precisam ser seguidas, conforme descrito a seguir.

3.2.3.1 Definir o espaço de nomes

O espaço de nomes do diretório indica a maneira como a informação será acessada. O modelo básico do espaço de nomes é um modelo em estrutura de árvore, que tradicionalmente reflete a estrutura organizacional e/ou geográfica da empresa.. Esta árvore é montada a partir do nome da entrada, seguindo a estrutura DN da entrada + RDN (Relative Distinguished Name) do pai. A figura 3.2 tem um exemplo de espaço de nomes, onde a consulta ao nome João Silva se dá através da sentença "c=BR, st=RS, o=Procergs, ou=Depto1, cn=João Silva":

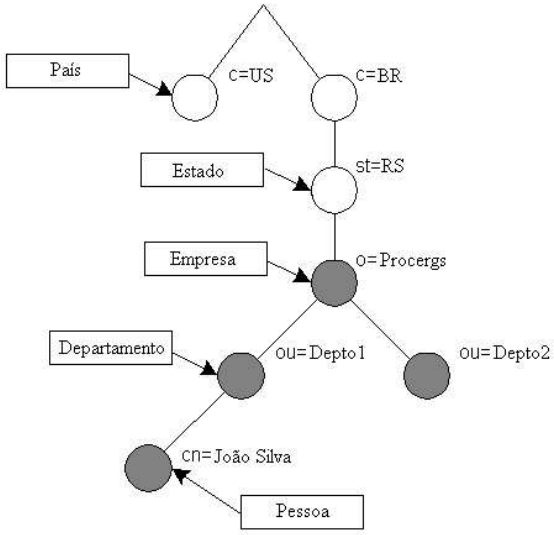

FIGURA 3.2 – Exemplo de Espaço de Nomes

Para uma definição correta do espaço de nomes, deve-se ter em mente os seguintes passos:

- verificar o número de entradas que o diretório implementará, prevendo o seu crescimento;
- considerar os tipos de entradas a serem armazenadas, decidindo se será permitido criar novos tipos de entradas no futuro;
- determinar se o diretório precisa ser centralizado ou distribuído;
- considerar se será preciso definir diferentes tipos de permissão para partes do diretório;
- verificar em qual parte dos dados será feita replicação;
- definir as aplicações que irão ter acesso ao diretório;

- decidir qual o atributo será usado como nome da entrada, garantindo sua unicidade;

3.2.3.2 Definir a topologia do diretório

A topologia descreve como o serviço de diretórios está distribuído entre os múltiplos servidores. Uma topologia bem definida é importante para:

- melhorar o desempenho;
- aumentar a disponibilidade;
- permitir um bom gerenciamento.

A figura 3.3 mostra um exemplo de topologia.

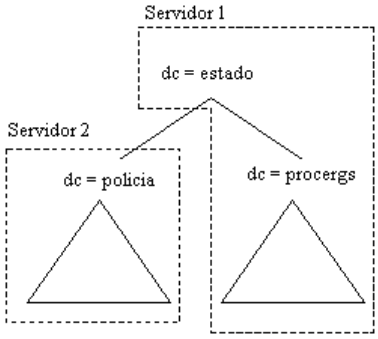

FIGURA 3.3 – um modelo de topologia

Na figura 3.3, a sub-árvore da Polícia encontra-se em um servidor diferente da sub-árvore da Procergs. Existem dois tipos de referência no modelo, que tornam o acesso transparente aos usuários:

- referência de conhecimento imediatamente superior: indica a partição pai da partição acessada;
- referências subordinadas: referências que apontam para partições filhas;

Um *referral* é uma informação retornada (informa o nome do servidor a ser contatado, o número da porta e o DN) por um servidor LDAP que indica ao cliente que outros servidores precisam ser contatados para que a requisição inicial seja preenchida.

3.2.3.3 Configurar o diretório

Para a definição de uma referência subordinada, é necessário criar um objeto do tipo *referral*, que deve pertencer a classe *referral* e deve ter o mesmo DN da sub-árvore que está no outro servidor. Geralmente, este objeto também tem uma classe auxiliar *extensibleObject*, para que a entrada possa conter um RDN apropriado.

Por exemplo, se o servidor1.policia.estado quer delegar a sub-árvore dc=policia,dc=estado para o servidor2.policia.estado, o seguinte objeto deve ser adicionado ao servidor1.policia.estado:

dn: dc=policia,dc=estado
objectClass: referral
objectClass: extensibleObject
dc: policia
ref: ldap://servidor2.policia.estado/dc=policia,dc=estado

3.2.3.4 Disponibilizar formas de acesso

O acesso ao diretório distribuído é feito de forma transparente, porém, é necessário informar se é desejado que o cliente acesse os *referrals* de forma automática. O suporte aos *referrals* só foram padronizados na especificação LDAPv3. A figura 3.4 mostra como o acesso a um servidor é feito, utilizando-se do *referral*.

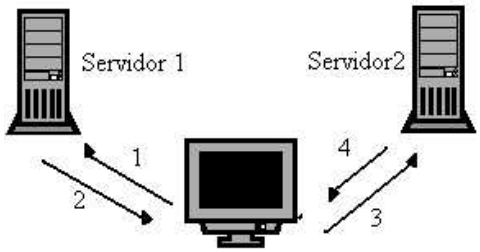

FIGURA 3.4 – Acesso ao servidor, utilizando referral

Os passos descritos na figura 3.4 são descritos como:

1 – cliente solicita uma informação;

2 – servidor 1 retorna uma referência ao servidor 2;

3 – cliente reenvia a solicitação ao servidor 2;

4 – servidor 2 retorna a informação ao cliente.

3.3 OpenLDAP

O projeto OpenLDAP é um esforço de colaboradores para desenvolver um conjunto de aplicações e ferramentas robusto, de nível comercial, bem caracterizado e de código fonte aberto, para o gerenciamento das informações armazenadas no LDAP. O projeto é gerenciado pela comunidade de voluntários do mundo todo que usam a Internet para comunicar-se, planejar e desenvolver o OpenLDAP Suite e sua documentação relacionada.

O software Direto já utiliza o OpenLDAP no seu projeto, que vai ser utilizado também para armazenar os dados referentes aos certificados e chaves públicas dos usuários.

3.4 OpenCA

O projeto OpenCA [COV2002] é um esforço de colaboradores para desenvolver uma Autoridade Certificadora robusta, bem-caracterizada, não-empacotada e de fonte-aberta, implementando os mais usados protocolos com o poder da criptografia para o mundo da WEB. Sua meta é a criação de um sistema de confiança centralizado, preparado para dar suporte a comunidade no desenvolvimento de uma infra-estrutura boa e barata.

O projeto OpenCA iniciou em 1998. A primeira idéia consistia na criação de três grandes partes – uma interface WEB em Perl, um *backend* em OpenSSL para as operações criptográficas e um banco de dados. Este conceito simples ainda é a base do OpenCA até o momento.

O sistema OpenCA é projetado para atender a uma infra-estrutura distribuída. Além de manipular uma AC *off-line* e uma AR *on-line*, pode-se criar uma hierarquia com três ou mais níveis, com a meta de obter uma flexibilidade maior. OpenCA não é um sistema completo trabalhando sozinho. Ele se utiliza de outros produtos de desenvolvedores da comunidade Open Source que são:

- Apache: O projeto Apache de um Servidor HTTP [APA2001] é um esforço de colaboradores para desenvolver e manter um servidor HTTP de fonte aberta, para sistemas operacionais modernos incluindo UNIX e Windows NT. A meta do projeto é desenvolver um servidor seguro, eficiente e extensível que proporcione serviços HTTP em sintonia com os padrões HTTP correntes;

- mod-ssl: mod_ssl [THE2001] é um módulo de segurança para o servidor Apache. O módulo mod_ssl usa as ferramentas fornecidas pelo projeto OpenSSL para adicionar uma característica muito importante ao Apache, a habilidade de cifrar as comunicações;

- OpenSSL: o projeto OpenSSL [OPE2004] inclui uma ferramenta que implementa o Secure Sockets Layer (SSL) e o Transport Layer Security (TLS) e é uma biblioteca de finalidade geral para criptografia. Hoje, o protocolo SSL é usado para a transmissão segura de dados. As ferramentas OpenSSL são usadas pelo módulo mod_ssl para fornecer segurança nas comunicações Web;

- OpenLDAP: veja na seção 3.3;

- Perl: Perl é uma linguagem de programação interpretada, com um grande número de características que a tornam muito útil para manipulação de dados textuais. Por isso é muito difundida para aplicações Web, como CGI, para servidores HTTP;

Componentes extras podem ser utilizados (um deles é o OCSP), porém, eles precisam ser compilados e instalados separadamente já que são pacotes de software que trabalham independentes do OpenCA.

A instalação do pacote completo consiste de vários passos incluindo compilações e configurações, definições de opções de interface, informações de parâmetros, de portas, de IPs, cópias de arquivos, entre outros.

As extensões de certificados são apenas citadas, mas não são documentadas.

As operações disponíveis para utilização pelo OpenCA podem ser definidos conforme abaixo:

- operações no módulo AC: a AC manipula dados da própria AC como obtenção de certificados da AC e gerenciamento da Lista de Certificados Revogados. A manipulação de dados do usuário se refere à requisição de certificados, criação de certificados, teste de certificados, revogação de certificados, consultas a listas de certificados válidos, revogados, expirados, suspensos e ainda permite a pesquisa de certificados. O módulo também controla as listas de requisições pendentes e listas de requisições de revogações pendentes;

- operações do módulo AR: a AR tem a função de administração do sistema, administrando as requisições pendentes, podendo editar, aprovar e assinar, aprovar e não assinar e excluir requisições. Manipula também os dados de requisições de revogações e mantém registros para informações sobre as operações realizadas. O módulo AR também manipula o LDAP, permitindo alterações, verificações de certificados da AC, visualização de certificados e visualização das LCRs.

Um exemplo de interface utilizada pelo projeto OpenCA pode ser vista na figura 3.5.

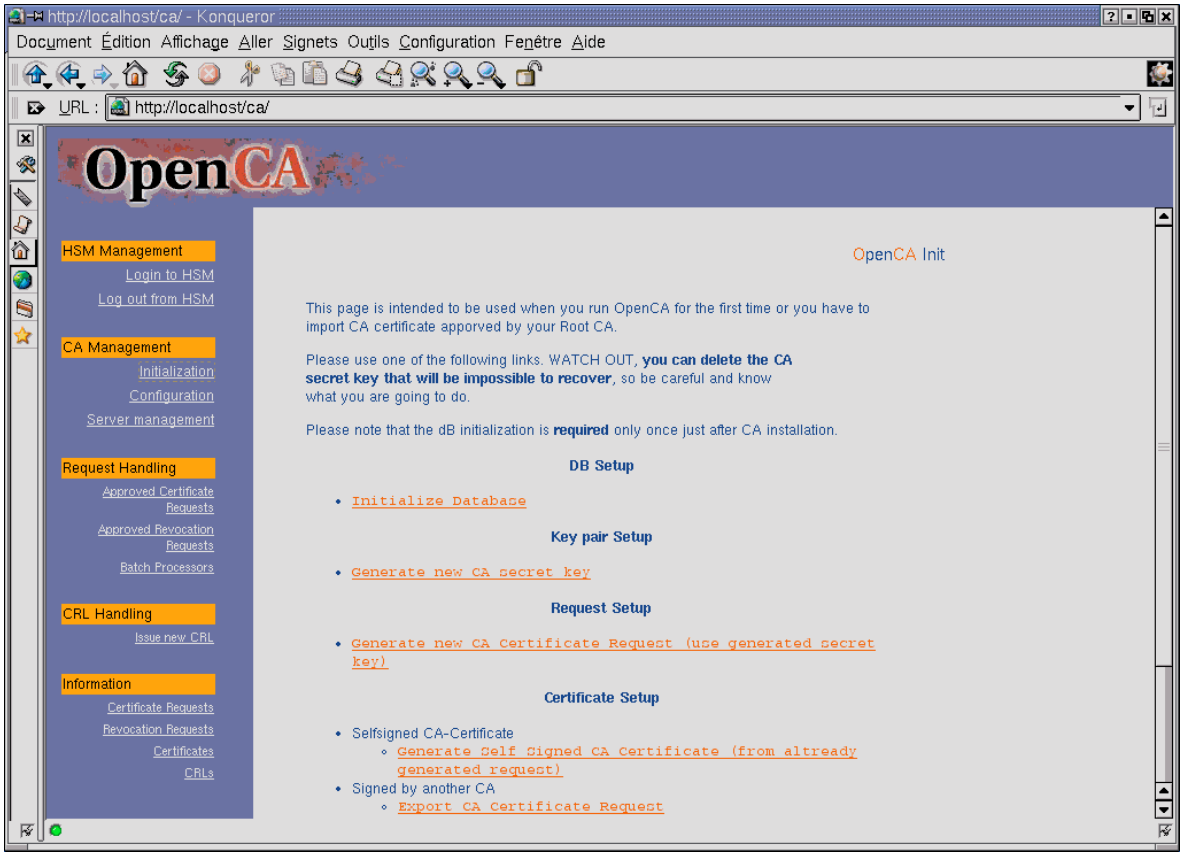

FIGURA 3.5 – exemplo de interface no projeto OpenCA

O projeto OpenCA é de difícil instalação. Os vários passos são confusos e cheios de detalhes. Não havia, até meados de setembro de 2003, nenhum manual ou instrução sobre a instalação ou mesmo sobre o uso do sistema. As interfaces do projeto são complexas, e necessitam de profissional capacitado para a instalação e configuração.

Nos teste realizados, foi feito uma requisição de certificado no módulo AR. Ao consultar a requisição no módulo AC, descobriu-se que ela desapareceu. A perda desta requisição deu total descrédito ao sistema.

Não existe uma definição clara de como os certificados revogados são criados e gerenciados.

A linguagem de programação é Perl, sendo necessária a reescrita de algumas partes (por exemplo, a requisição de certificados através do software Direto para que ele pudesse ser acoplado ao Direto).

Embora o OpenCA não tenha sido utilizado no desenvolvimento da Infra-estrutura de chaves públicas para uso pelo Direto, seu estudo serviu de base e deu idéias para o desenvolvimento das rotinas necessárias para o gerenciamento das chaves e certificados dos usuários.

3.5 OpenSSL

O protocolo SSL foi criado com o objetivo de proporcionar mecanismos de autenticação e sigilo entre duas aplicações que se comunicam via algum protocolo de comunicação. Outros aspectos importantes considerados no momento de sua concepção foram: interoperabilidade, permitindo a comunicação com outra aplicação sem que haja a necessidade de entrar em detalhes a respeito de sua implementação, extensiblidade, que permite criar novas rotinas e funcionalidades baseadas em mecanismos pré-existentes do protocolo e, por fim, eficiência, tornando o protocolo viável para o uso entre aplicações cliente-servidor via Internet.

A arquitetura do SSL é disposta em camadas, a exemplo do TCP/IP. Uma delas, a chamada Record Layer, recebe informações não cifradas das aplicações, dispondo-as em blocos numerados seqüencialmente. Estes blocos então passam por uma compactação, seguida da geração de códigos de autenticação (MACs). Em seguida, os blocos são cifrados e enviados. A numeração das mensagens enviadas é importante para facilitar o trabalho do receptor na detecção de blocos em falta, alterados ou injetados por terceiros.

O projeto OpenSSL é um esforço conjunto para se desenvolver um conjunto de ferramentas robusto, de qualidade, completo em funcionalidades e de código aberto, que implementa o protocolo SSL e vários algoritmos e primitivas criptográficas de uso comum, incluindo algoritmos de troca de chaves, funções de hash, algoritmos simétricos e assimétricos. Este conjunto de ferramentas se apresenta na forma de duas bibliotecas e um conjunto de programas que implementam as rotinas por elas disponibilizadas. Os mecanismos do SSL estão implementadas na *libssl*, e os outros algoritmos estão implementados na *libcrypto*.

O Direto já utiliza OpenSSL na sua implementação.

As bibliotecas do OpenSSL são implementações para C e C++. A utilização destas bibliotecas junto ao software Direto faria com que ele perdesse uma das suas características: a portabilidade da aplicação.

3.6 Pacote JDK 1.4.2 da Sun Microsystems

A versão 1.4 [SUN2002] provê suporte adicional em padrões tecnológicos de indústria como XML, DOM, SSL, Kerberos, LDAP e CORBA para garantir operabilidade sobre plataformas, sistemas e ambientes heterogêneos.

Três pacotes de segurança que antes eram opcionais, agora são incluídos como parte da plataforma J2SE. São eles:

- Java Authentication and Authorization Service permitem aos desenvolvedores autenticar usuários e a garantir controle de acesso baseado em usuários, grupos ou regras;

- Java Secure Socket Extension (JSSE) implementa uma versão Java dos protocolos SSL e TLS e inclui funcionalidades para criptografia de dados, autenticação de servidores, integridade de mensagens e autenticação opcional de clientes para ajudar a proteger a privacidade e integridade dos dados e sua transferência sobre a rede;

- Java Cryptography Extension (JCE) provê a implementação de um ambiente para criptografia, e algoritmos para geração e autorização através de chaves e de Message Authentication Code (MAC).

A linguagem Java é a linguagem padrão de desenvolvimento do software Direto. As características de segurança e os métodos de criptografia e assinatura disponíveis atendem aos requisitos necessários à implementação da ICP da Procergs.

A utilização do pacote JDK 1.4.2 será o utilizado para o desenvolvimento do protótipo, implementando os métodos de segurança disponíveis.

3.7 KeyTool

Keytool [SUN2004] é uma ferramenta Java para gerenciar chaves e certificados. Ela armazena chaves e certificados em *keystore*s. Os *keystores* são arquivos onde a chave privada é protegida por senha.

Keytool permite que usuários administrem seus próprios pares de chaves e os certificados associados a eles. Eles podem também armazenar certificados de chaves públicas (que contém as chaves públicas) de outras entidades, com as quais desejam comunicar-se. Podem existir dois tipos de entrada em um *keystor*e: para as chaves, o que inclui a chave privada e o certificado contendo a chave pública de uma determinada entidade, e para certificados (contendo a chave pública de outras entidades) confiáveis. A entrada para as chaves armazena as informações de forma secreta em um formato protegido, para prevenir acesso não autorizado. A outra entrada armazena certificados confiáveis. Um certificado é dito confiável porque o possuidor do *keystore* confia na chave pública daquele certificado.

Essa entrada é necessária caso o possuidor do *keystore* deseje receber dados/aplicações de outras entidades e autenticá-las. Os dois tipos de entrada do *keystore* estão associadas a um *alia*s. É o *alias* que identifica o possuidor de chaves e certificados confiáveis no *keystor*e, como ilustra a figura 3.6.

Um *keystore* pode conter vários *alia*s. O *alias* é um nome utilizado para identificar o possuidor de chaves e certificados no *keystor*e. Por exemplo, se existe um *alias* chamado *superuser*s, a primeira entrada armazena a chave privada e o certificado contendo a chave pública de *superuser*s. Para acessar as chaves, deve-se fornecer o *alia*s. A segunda entrada armazena certificados contendo a chave pública de outras entidades confiáveis para *superuser*s, com as quais *superusers* provavelmente irá se comunicar no futuro.

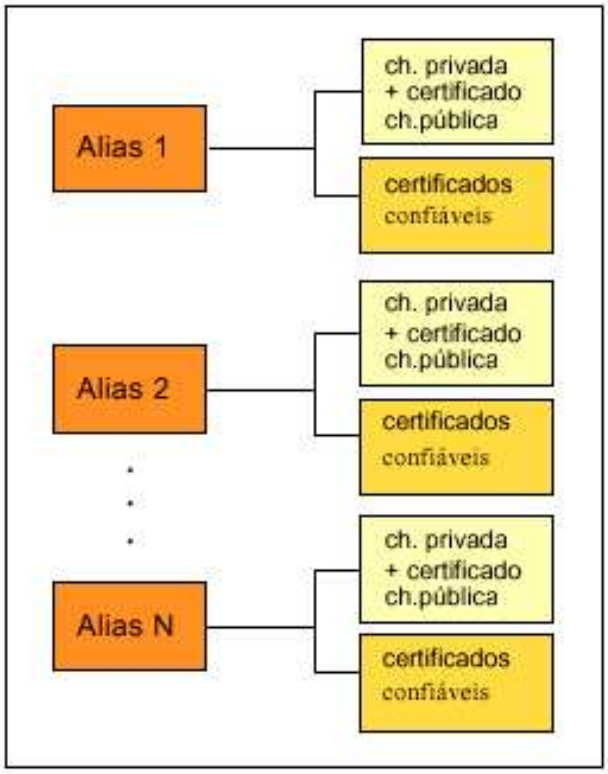

FIGURA 3.6 – Estruturação do *keystore*

Os certificados podem ser exportados e importados do *keystor*e. Exportar um certificado significa extraí-lo do *keystore* para que ele possa ser enviado a uma entidade que precisará autenticar uma chave pública. A entidade que recebeu o certificado exportado deve, então, entrar em contato com o possuidor do certificado e verificar sua autenticidade comparando os *fingerprints* (ou *hashe*s) do certificado original com os gerados a partir do certificado recebido. Se os *fingerprints* forem iguais significa que o certificado é válido, então basta importá-lo, ou seja, adicioná-lo à lista de certificados confiáveis do *keystor*e.

A ferramenta *keytool* gera os certificados auto-assinado, ou seja, o próprio possuidor da chave privada garante a autenticidade do certificado contendo a chave pública.

Keytool possibilita a criação de uma requisição de assinatura para certificado, ou seja, gera-se um arquivo, que será enviado para uma entidade certificadora, contendo uma requisição de assinatura. Para gerar a requisição, já deve existir no *keystore* a chave privada e o certificado *auto-assinado*. A *Keytool* gera a requisição a partir desses dados

armazenados. Estando de posse do arquivo para requisição, deve-se enviá-lo para uma entidade certificadora, que irá assinar o certificado e retorná-lo.

Tendo um certificado com a chave pública autenticada por uma autoridade certificadora, basta substituir o certificado auto-assinado do *keystore* pelo novo certificado. Os certificados gerados pela ferramenta *keytool* estão no padrão X.509.

A ferramenta keytool não realiza a assinatura de certificados digitais, que é uma das metas principais da ICP da Procergs.

Keytool é uma interface para a classe *Keystore*, sendo realizada através de linhas de comando. A correta execução de seus comandos necessitaram de chamadas extras para testes da criação das chaves e certificados.

Os *keystores* são armazenados em arquivos, no próprio disco, e precisariam ser manipulados para a correta gravação no LDAP e no PostgreSQL.

Por ser uma ferramenta que gera dados confiáveis, esta ferramenta será utilizada para criação de requisições e certificados para testes junto ao protótipo implementado da ICP.

3.8 Bouncy Castle

O pacote de Criptografia Bouncy Castle é uma implementação Java de algoritmos criptográficos, desenvolvido pela "Legião do Castelo Tremulante" (Legion of the Bouncy Castle), anteriormente OpenJCE.org. Este software é de código livre, com licença de distribuição baseada na licença do Consórcio X do MIT.

As APIs Bouncy Castle Crypto consistem no seguinte:

- uma API simples (*lightweight*) em Java que implementa os mais comuns algoritmos de criptografia como DES, Blowfish e IDEA;
- um provedor para JCE e JCA;
- uma implementação "clean room" para a JCE 1.2.1;
- geradores para certificados X.509 Versão 1 e Versão 3 e arquivos PKCS#12;
- geradores para S/MIME e CMS (PKCS#7);
- uma versão jar assinada, própria para JDK 1.4 e a Sun JCE.

O pacote JDK 1.4.2 da Sun não assina certificados (apenas certificados auto-assinados) e não provê uma infra-estrutura clara para suporte a criação e leitura de e-mails cifrados e/ou assinados.

A classe Boucy Castle, incorporada ao pacote Java irá prover todas as funcionalidades que o projeto de Infra-estruturas de chaves públicas do Direto pretende desenvolver.

3.9 Representação de Dados

A complexidade dos dados a serem manipulados e a diferença de representação destes dados nas diversas máquinas existentes levaram a criação de uma forma de representação de dados mais eficiente. A abstração de dados [BUR2002] permite que uma parte do sistema seja especificada sem se preocupar com a maneira como essa parte será realmente implementada ou representada.

A construção da ICP para o Direto precisa prover abstração de dados, já que necessitará, com o passar do tempo, trocar informações com outras infra-estruturas desenvolvidas com outras ferramentas. A única maneira de manter o correta leitura e entendimento dos dados de um certificado por produtos diferentes é utilizando as notações abstratas de objetos. Os métodos utilizados para especificar os objetos de forma abstrata serão apresentadas à seguir.

3.9.1 ASN.1

A Abstract Syntax Notation 1 (ASN.1 ou Sintaxe de Notação Abstrata 1) [BUR2002] define uma notação para especificar valores e definir tipos. A definição de um tipo consiste numa coleção de campos que, no mais baixo nível, consiste de um identificador, uma possível etiqueta (rótulo ou "tag"), uma referência e uma possível indicação de que aquele campo é opcional (pode ser omitido). A ASN.1 tem quatro tipos definidos:

- tipos simples: que são atômicos e não tem nenhum componente;
- tipos estruturados: que tem componentes;
- tipos marcados com tags (*tagged types*): que são derivados de outros tipos;
- Outros tipos: incluem o tipo CHOICE e o tipo ANY.

Tipos e valores de ASN.1 são expressos em uma notação flexível como uma linguagem de programação, com as seguintes regras especiais:

- layout não é importante;
- comentários são delimitados por pares de hífens (-) ;
- identificadores (nomes de valores e de campos) e referências de tipo (nome de tipos) consistem em letras maiúsculas e minúsculas, dígitos, hífens e espaços; os

identificadores começam com letras minúsculas; referências de tipo começam com letras maiúsculas;

3.9.2 BER

As Basic Encoding Rules (BER ou Regras Básicas de Codificação) para a ASN.1 fornecem uma ou mais maneiras de representar qualquer valor da ASN.1. BER provê um algoritmo que especifica como um valor de qualquer estrutura (tipo) definida usando ASN.1 deve ser codificada para transmissão. O uso do algoritmo no sentido contrário permite que qualquer receptor que tinha conhecimento da definição do tipo ASN.1, possa decodificar os bits que chegam em um valor daquele tipo.

A codificação de um tipo de dados ASN.1, usando as Basic Encoding Rules, permite que um receptor, sem conhecimento da definição de tipo, reconheça o inicio e o fim das construções (SEQUENCE, SET, etc) e os octetos representando os tipos básicos de dados (BOOLEAN, INTEGER). No uso mais simples da notação, e também possível determinar, a partir da codificação, as construções efetivamente usadas e os tipos de dados básicos) .

A codificação BER consiste de quatro componentes que deverão aparecer na seguinte ordem

- octetos identificadores: identificam a classe e o número de tag do valor ASN.1 e indica se o método é primitivo ou construído;

- octetos de comprimento: para os métodos de comprimento definido, fornecem o número de octetos de conteúdo. Para os métodos constituídos de comprimento indefinido, indicam que o comprimento é indefinido;

- octetos de conteúdo: Zero ou mais octetos codificando os valores sendo transmitidos;

- octetos de final de conteúdo: Dois octetos zero. Este campo somente estará presente quando o comprimento do conteúdo não for conhecido ao ser iniciada sua transmissão; neste caso, no octeto de comprimento será sinalizada esta forma de delimitação de conteúdo.

3.9.3 DER

Distinguished Enconding Rules (DER ou Regras de Codificação Diferenciada) é um subconjunto de BER e fornece uma maneira única de representar qualquer valor ASN.1 como um string de octetos. O DER foi concebido para aplicativos nos quais uma única codificação de string de octetos é necessária, como no caso quando uma assinatura digital é computada com um valor de ASN.1.

3.10 Padrões

PKCS (Public-Key Cryptography Standards) é um conjunto de padrões para criptografia de chave pública, desenvolvida pela RSA Laboratories em cooperação com um consórcio informal de empresas, que originalmente incluía a Apple, a Microsoft, a DEC, a Lotus e a Sun.

PKCS já é um padrão de fato em criptografia e continua a ser desenvolvido, para facilitar a interoperabilidade entre metodologias de criptografias de chaves públicas. Visando a interoperabilidade da troca de certificados pelo software Direto, os padrões descritos nas próximas seções serão utilizados no desenvolvimento do protótipo.

3.10.1 PKCS #8 - Private-Key Information Syntax Standard

O padrão PKCS#8 descreve uma sintaxe para informações de chave privada. Estas informações incluem uma chave privada para algum algoritmo de chave pública e um conjunto de algoritmos.

A intenção de incluir um conjunto de atributos é prover uma forma simples de usuários obterem alguma confiança na informação como uma DN ou a chave pública da autoridade certificadora raiz.

A sintaxe é descrita por:

```
PrivateKeyInfo ::= SEQUENCE      {
        version Version,
        privateKeyAlgorithm PrivateKeyAlgorithmIdentifier,
        privateKey PrivateKey,
        attributes [0] IMPLICIT Attributes OPTIONAL
}

Version ::= INTEGER
PrivateKeyAlgorithmIdentifier ::= AlgorithmIdentifier
PrivateKey ::= OCTET STRING
Attributes ::= SET OF Attribute
```

O valores de cada campo podem ser descritos como:

- Version: é o número da versão, para compatibilidade com revisões futuras no padrão. Para o padrão atual, deve-se utilizar 0 (zero);

- PrivateKeyAlgorithmIdentifier: identifica o algoritmo de chave pública;

- PrivateKey: é o valor da chave privada;

- Attributes: é um conjunto de atributos, que são informações estendidas e que serão cifradas junto com a chave privada.

3.10.2 PKCS #10 - Certification Request Syntax Standard

Uma requisição de certificados consiste de três partes: informações da requisição do certificado, identificador do algoritmo de assinatura e a assinatura digital.

O processo na qual uma requisição de certificado é construído equivale aos seguintes passos:

1- um CertificationRequestInfo, contendo o DN do sujeito, a chave publica do sujeito e, opcionalmente, um conjunto de atributos, é construído por uma entidade requisitante de certificados;

2- o CertificationRequestInfo é assinado com a chave privada do certificado;

3- o valor CertificationRequestInfo, o identificador do algoritmo de assinatura e a assinatura da entidade são agrupados em um valor chamado CertificationRequest.

Uma autoridade aceita a requisição autenticando a entidade requisitante e verificando a assinatura da entidade, e , se a requisição é válida, construindo um certificado X.509. Se a requisição contém algum atributo, a autoridade certificadora pode usar estes valores, bem como outros valores conhecidos, para construir um certificado X.509 com extensões.

A sintaxe da CertificationRequestInfo é descrita por:

```
CertificationRequestInfo ::= SEQUENCE {
    version          INTEGER,
    subject          Name,
    subjectPKInfo    SubjectPublicKeyInfo{{ PKInfoAlgorithms }},
    attributes       [0] Attributes{{ CRLAttributes }}
}

SubjectPublicKeyInfo { ALGORITHM : IOSet} ::= SEQUENCE {
    algorithm        AlgorithmIdentifier {{IOSet}},
    subjectPublicKey BIT STRING
}

PKInfoAlgorithms ALGORITHM ::= {
    Algorithm,

   Attributes { ATTRIBUTE:IOSet } ::= SET OF Attribute{{ IOSet }}

CRIAttributes  ATTRIBUTE  ::= {
    atributos }
```

```
Attribute { ATTRIBUTE:IOSet } ::= SEQUENCE {
    Type    ATTRIBUTE.&id({IOSet}),
    values  SET SIZE(1..MAX) OF ATTRIBUTE.&Type({IOSet}{@type})
}
```

O valores de cada campo podem ser descritos como:

- version: é o número da versão, para compatibilidade com futuras revisões do padrão. Nesta versão, o utilizado é zero;
- subject: é o Distinguished Name (DN) do sujeito do certificado
- subjectPublicKeyInfo: contém informações sobre a chave pública do certificado. A informação identifica o algoritmo de chave pública da entidade e a própria chave pública da entidade.
- attributes: é uma coleção de atributos que provêm informações adicionais sobre o sujeito do certificado.

A sintaxe da CertificationRequest é descrita por:

```
CertificationRequest ::= SEQUENCE {
    certificationRequestInfo  CertificationRequestInfo,
    signatureAlgorithm        AlgorithmIdentifier{{ SignatureAlgorithms }},
    signature                 BIT STRING
}
AlgorithmIdentifier {ALGORITHM:IOSet } ::= SEQUENCE {
    algorithm       ALGORITHM.&id({IOSet}),
    parameters      ALGORITHM.&Type({IOSet}{@algorithm}) OPTIONAL
}
SignatureAlgorithms ALGORITHM ::= {
    ... valores conhecidos são colocados aqui }
```

O valores de cada campo podem ser descritos como:

- CertificateRequstInfo: é a informação da requisição do certificado;
- SignatureAlgorithm: identifica o algoritmo de assinatura e quaisquer parâmetros associados sobre o qual a requisição é assinada;
- Signature: é o resultado da assinatura da informação da requisição com a chave privada do sujeito que requisita o certificado.

O processo de assinatura consiste de dois passos:

1 – o valor de certificationRequestInfo é codificado em DER gerando um octet string;

2 – o resultado do passo 1 é assinado com a chave privada do sujeito que requisita o certificado com o algoritmo de assinatura específico, gerando um bit string, a assinatura.

3.10.3 PKCS #5 - Password-Based Cryptography Standard

Em muitas aplicações de criptografia de chave pública a segurança do usuário depende basicamente de um ou mais valores de texto secretos ou senhas. Desde que uma senha não é diretamente aplicável como uma chave em muitos criptosistemas convencionais algum processamento da senha é necessário para realizar operações criptográficas com ela. Como senhas são geralmente pequenas, um cuidado especial é necessário para defender-se contra ataques de pesquisa.

Um método de criptografia baseada em senha é a combinação de uma senha com um *salt* para produzir uma chave. Um *salt* pode ser visto como um índice para um grande conjunto de chaves derivadas da senha, e que não necessita ser mantida secreta para ser segura.

Mesmo que seja possível para um oponente construir uma tabela de possíveis senhas (também chamada de "dicionário de ataque"), construir uma tabela de possíveis chaves deve ser difícil, desde que existam muitas chaves possíveis para cada senha. Um oponente é então limitado a pesquisar todas as senhas separadamente para cada *salt.*

Outro caminho para a criptografia baseada em senha é a construção de técnicas de derivação de chaves com um custo relativamente alto, já que incrementa o custo das pesquisas exaustivas. Uma forma conhecida é incluir um contador de repetições na técnica de derivação de chaves, indicando quantas vezes deve-se repetir alguma função pela qual as chaves serão derivadas.

3.10.3.1 Salt

Um *salt* na criptografia baseada em senha tem tradicionalmente servido a proposta de produzir um grande conjunto de chaves correspondendo a uma dada senha, e pode ser visto como um dado adicional concatenado a senha. Uma chave individual de um conjunto de chaves é selecionada aplicando a função de derivação de chaves KDF, como *DK = KDF(P,S)*, onde DK é a chave derivada, P é a senha e S é o *Salt*.

Os benefícios do uso do *Salt* pode ser descritos como:

- é difícil para um oponente pré-computar todas as chaves ou os pares de chaves mais prováveis correspondentes a um dicionário de senhas. Se o *salt* é de 64 bits, por instância, devem existir 2^{64} chaves para cada senha.

- é improvável que a mesma senha seja selecionada duas vezes.

Um exemplo do uso de salt, considerando um *salt* de apenas 2 bits, pode ser entendido utilizando o exemplo abaixo, que considera um dicionário de ataque com duas chaves possíveis baseadas em senhas comuns:

md5(‘Marcia’)
md5(´Pedrini´)

Quando o atacante encontra algum dado cifrado com senha, e pode simplesmente usar as duas chaves pré-criadas do dicionário para decifrar os dados e ver se uma resposta correta é criado. Agora, se um salt for utilizado o atacante deve incluir estas possibilidades:

md5(0,‘Marcia’)
md5(0,‘Pedrini’)

md5(1,‘Marcia’)
md5(1,‘Pedrini’)

md5(2,‘Marcia’)
md5(2,‘Pedrini’)

md5(3,‘Marcia’)
md5(3,‘Pedrini’)

3.10.3.2 Contador de Repetições

Um contador de repetições tem tradicionalmente servido a proposta de aumento do custo de produção de chaves com uma senha, assim, também incrementando a dificuldade de um ataque. Um contador de repetições especifica quantas vezes uma senha deve ser resumida para produzir uma chave.

O exemplo a seguir mostra um dicionário com entradas para um, dois e três contadores de repetição:

md5(‘Marcia’)
md5(md5(‘Marcia’))
md5(md5(md5(‘Marcia’)))

3.10.3.3 Funções de derivação de chaves

Uma função de derivação de chaves (KDF) produz uma chave derivada de uma chave base e outros parâmetros. Em uma função de derivação de chaves baseada em senha, a chave base, uma senha e os outros parâmetros são um valor de *salt* e um contador de repetições.

A criptografia de uma mensagem deve envolver os seguintes passos

- selecionar um salt S e um contador de repetições C;
- selecionar um tamanho em *octetos* para a chave derivada dkLen;
- aplicar a função de derivação de chaves para a senha, o *salt*, o contador de repetições e o tamanho de chave para produzir uma chave derivada *DK = KDF(P, S, c, dkLen);*
- cifrar a mensagem M com um algoritmo de criptografia básico com a chave DK para produzir um texto cifrado C.

A decifragem da mensagem deve envolver os seguintes passos:

- obter o *salt* S da operação;
- obter o contador de repetições c para a função de derivação de chaves;
- obter o tamanho da chave em octetos, dkLen, para a chave derivada do esquema de criptografia básico;
- aplicar a função de derivação de chave para a senha P, o *salt* e o contador de repetições c para produzir a chave derivada DK, de tamanho dkLen;
- decifrar a mensagem C com o esquema de criptografia básico sobre a chave derivada DK para recuperar uma mensagem M.

3.11 Considerações Finais

Neste capítulo foram apresentadas as ferramentas, as tecnologias, os padrões e as representações de dados que serão utilizadas no decorrer do desenvolvimento deste trabalho.

Com este estudo pôde-se ter uma idéia da complexidade e do esforço que será empregado para a conclusão do objetivo principal, que é manter a segurança das mensagens trafegadas pelo software Direto.

4 Direto

Neste capítulo será feita uma explanação sobre o projeto Direto [PRO2000], com uma breve descrição das tecnologias e ferramentas utilizadas para o seu desenvolvimento.

4.1 Projeto Direto

No início do ano de 1999, a PROCERGS (Companhia de Processamento de Dados do Rio Grande do Sul) [BAL2002] identificou uma falta de padronização nas ferramentas de correio eletrônico e *workflow* utilizadas pelo Estado do Rio Grande do Sul, bem como algumas dificuldades e problemas com o uso das ferramentas adotadas pela maioria dos seus clientes, o MEMO e o Notes. Os pontos mais críticos destas ferramentas eram:

- falta de uma ferramenta padrão de correio eletrônico para o Governo do Estado que facilitasse a comunicação interna e externa;
- convivência de duas ferramentas de correio eletrônico adotadas pelo Estado (Memo e Notes), uma com interface caractere e outra com interface gráfica;
- ferramentas de correio eletrônico existentes estavam sub-implementadas, pois alguns recursos do Notes como replicação, trabalho *off-line* e acesso remoto não foram implementados por problemas técnicos de segurança. No Memo, alguns recursos também não foram implementados;
- falta de conhecimento dos usuários das ferramentas utilizadas;
- o correio eletrônico Memo é visto como ferramenta com obsolescência tecnológica pelo tipo de interface que apresenta;
- a montagem de uma infra-estrutura de hardware e software para o Estado depende da disponibilidade de recursos financeiros elevados;
- familiaridade dos usuários do Estado com correio, Internet e ferramentas de correio eletrônico como Exchange, Outlook, e Eudora;
- falta de integração entre as agendas do Memo e do Notes.

Com o propósito de resolver esta situação foi criado um grupo de trabalho que tinha como objetivo principal realizar um estudo amplo (técnico, operacional, financeiro, entre outros), das opções de mercado na área de correio eletrônico e *workflow*, e a apresentação de um relatório técnico que permitisse a tomada de decisão sobre qual a melhor alternativa a ser adotada, levando em conta o tempo, os custos e a qualificação técnica do pessoal envolvido, para a substituição das ferramentas em uso.

A partir dos estudos realizados, foi criado um novo grupo, para atender uma nova demanda que surgiu: a de se avaliar uma solução aberta, de código fonte livre, com a utilização de protocolos padrão Internet. Os objetivos deste grupo eram:

- verificar a possibilidade de implantar uma solução aberta de correio eletrônico, agenda, diretório e *workflow;*
- utilização de protocolos e padrões Internet de domínio público;
- implementar um protótipo para ser avaliado junto com outros protótipos de fornecedores;
- aumentar o conhecimento da empresa em plataformas e protocolos abertos.

O grupo seguiu alguns pressupostos básicos que orientaram o planejamento do desenvolvimento:

- prover independência de plataforma no lado cliente;
- prover independência de plataforma no lado servidor;
- prover independência de gerenciador de banco de dados;
- prover independência de servidor Web;
- baixo custo;
- ser modular, permitindo a inclusão de novas funções no futuro.

Na avaliação dos itens para a solução, e do estudo de várias alternativas técnicas, foi escolhida uma solução que apresenta como estrutura de desenvolvimento a utilização de *servlets* com:

- acesso a banco de dados PostGres via JDBC;
- utilização de navegadores (HTML e Javascript para validação de campos);
- utilização de servidores Intel com sistema operacional FreeBSD;
- servidor de páginas Web Apache (pela sua performance e baixo custo).

De acordo com a solução proposta, os serviços implementados foram:

- Serviço de Correio

  - protoloco IMAP para serviço de acesso a mensagens com a implementação Cyrus 1.5.19;

  - protoloco SMTP para serviço de entrega de mensagens com implementação SendMail 8;

  - disponibilização de uma aplicação (*servlet*) para acesso via navegador.

- Serviço de Diretório

  - protocolo LDAP com a implementação OpenLDAP;

  - disponibilização de uma aplicação (*servlet*) para acesso via navegador.

- Serviço de Agenda

  - desenvolvimento de uma aplicação Java (*servlet*) com acesso via navegador e com dados armazenados no banco da dados. Os campos e tabelas definidos na aplicação são sintática e semanticamente iguais aos campos propostos no formato vCalendar. Para permitir a interoperação com produtos de mercado que implementem o formato vCalendar foi proposto o desenvolvimento de uma operação de exportação e importação desse formato na aplicação.

- Serviço de *Workflow*

  - a alternativa selecionada foi o desenvolvimento de uma aplicação segundo a estrutura proposta (*servlets*) com forte integração com outros serviços propostos no projeto.

4.2 O produto Direto

O Direto é um produto de correio, agenda e catálogo com acesso pela Internet, baseado em software livre, o que diminui o custo do projeto e possibilita o seu uso por organizações interessadas sem custo algum, com independência de plataforma e flexibilidade de aperfeiçoamento.

O Direto possui interface Web para ser acessado a partir de um navegador, podendo ser acessado de qualquer ponto da Web sem a necessidade de instalação de um cliente ou de alguma configuração específica.

4.2.1 Tecnologias Utilizadas

Neste capítulo serão descritas rapidamente as principais tecnologias utilizadas para o desenvolvimento do software Direto.

4.2.1.1 Sistema Operacional

O Direto, por seguir a filosofia de software livre, optou por ser executado sobre o sistema operacional Linux, um dos precursores da filosofia de software livre.

A escolha pelo Linux não se deu apenas pelo fato desse sistema operacional ser um software livre, mas porque ele está entre os mais robustos e confiáveis sistemas operacionais existentes no mercado, fato comprovado pela sua ampla utilização.

Linux possui características que permitem sua execução na grande maioria das plataformas de hardware existentes. Esta portabilidade é de grande relevância do ponto de vista da arquitetura do Direto.

4.2.1.2 Linguagem de Programação

A linguagem de programação utilizado no desenvolvimento do Direto é o Java, uma linguagem de livre distribuição criada com o propósito principal de servir ao desenvolvimento de software para redes de computadores.

Dentre as principais características da linguagem Java, podemos citar:

- orientação a objetos: o paradigma de orientação a objetos torna a programação modular e de fácil compreensão, além de aumentar a produtividade da reutilização de objetos;

- portabilidade: o conceito de portabilidade sob a qual foi desenvolvida a linguagem de programação Java é de que sistemas de computação devem ser escritos uma única vez e podem ser executados em diferentes tipos de plataforma;

- robustez: presença de alguns recursos como o *garbage colector* (coleta automática de lixo) e a presença de mecanismos de tratamento de exceções tornam as aplicações mais robustas.

4.2.1.3 Sistema gerenciador de banco de dados

O Direto utiliza-se dos recursos do banco de dados PostgreSQL, que é um Sistema gerenciador de banco de dados objeto-relacional, que implementa a maioria dos comandos SQL, incluindo sub-seleções, transações, funções e tipos definidos pelo usuário.

4.2.1.4 *Servlet*

Os *servlets* foram integrados a linguagem Java como pequenas aplicações baseadas em Java para adicionar funcionalidades dinâmicas a servidores Web. Esta solução, além de prover independências de fornecedor, tem como principais características:

- eficiência: o código de inicialização de um *servlet* é executado somente a primeira vez que o servidor Web o carrega;

- persistência: os *servlets* podem manter estado entre requisições. Quando um *servlet* é carregado ele permanece na memória do servidor enquanto estiver recebendo requisições;

- robustez: por serem desenvolvidos em Java herdam suas características de robustez, consistência e manipulação eficiente de erros;

- adaptação natural a Internet;

- extensibilidade: possuem as características de herança e polimorfismo, permitindo reutilização de código;

4.2.1.5 JavaServer Pages (JSP)

JSP [SUN2001] é uma tecnologia baseada em Java que simplifica o processo de desenvolvimento de sites dinâmicos. JSP é composto de *tags* que são incluídas junto ao código HTML para serem executadas durante uma requisição. O código JSP é compilado para *servlets* Java, garantindo melhor desempenho do que linguagens *script* interpretadas.

4.2.1.6 eXtensible Markup Language (XML)

XML [XML2001] é um padrão para publicação, combinação e intercâmbio de documentos multimídia. XML lida com instruções embutidas no corpo de documentos, chamadas de *tags*, que permitem a descrição de dados. XML tem como base linguagens mais antigas como SGML e HTML, sendo atualmente empregada na representação de estruturas de dados estruturados e semi-estruturados e seu intercâmbio na Web.

4.2.1.7 eXtensible Stylesheet Language (XSL)

A linguagem XSL é utilizada para acrescentar aspectos de apresentação aos elementos de um documento XML. Desta forma, é possível criar múltiplas representações da mesma informação a partir de vários documentos XSL diferentes aplicados a um único documento XML. Um documento XSL pode conter uma série de regras, denominadas *templates*.

4.2.2 Arquitetura do Direto

O Direto está dividido em componentes de acordo com suas responsabilidades. Isto facilita a alteração e atualização de determinado componente e também sua distribuição. A figura 4.1 apresenta a arquitetura do Direto.

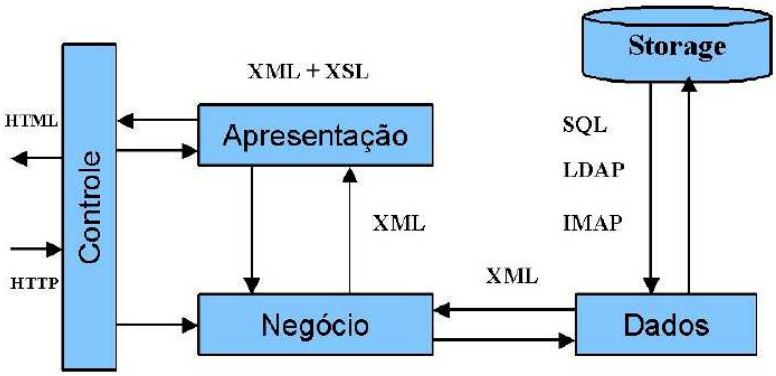

FIGURA 4.1 – Arquitetura do Direto

4.2.2.1 Controle

O componente de controle é quem recebe as requisições dos usuários e possui como responsabilidades autenticação de usuários, controle de sessão e *cache* de sessões de usuários.

4.2.2.2 Negócio

O componente é constituído de classes Java que se comunicam com outros serviços para realizar determinada tarefa. A comunicação com os serviços é feita utilizando-se APIs da linguagem Java. As principais APIs utilizadas são: JavaMail, JNDI, JDBC.

4.2.2.3 Dados

O Direto utiliza diversas fontes de dados para armazenar os dados dos usuários, porque os diferentes tipos de serviços que ele utiliza necessitam de diferentes tipos de sistemas de armazenamento, dependendo da natureza do serviço. Os dados do Direto estão armazenados em: Banco de Dados Relacional, IMAP e LDAP.

4.2.2.4 Apresentação

Esta camada é responsável por receber os dados da camada de negócios e formatar os dados para serem enviados para o navegador do usuário. Os dados são recebidos em formato XML e são transformados para HTML utilizando-se um *template* XSL.

4.3 Projeto Direto / Fapergs

O Grupo de Processamento Paralelo e Distribuído (GPPD – www.gppd.inf.ufrgs.br do II-UFRGS, juntamente com grupos de pesquisa das universidades regionais Lasalle (Canoas), UCPel (Pelotas) e URCAMP (Bagé), desenvolveram inúmeras melhorias para o Direto no contexto de um projeto de cooperação com a Procergs financiado pela Fapergs

(http://www.inf.ufrgs.br/procpar/direto/), onde a lista de atividades e realizações pode ser analisada na página (http://www.inf.ufrgs.br/procpar/direto/trabalhos.html). Os temas principais dos trabalhos dessa parceria foram: aumento de desempenho nas funções de negócio Java, especificação em UML, catálogo de usuários distribuído e replicado, reengenharia de software do módulo agenda com transações, modelagem J2EE dos principais módulos e outros.

5 Modelo Geral

Neste capítulo será apresentada uma solução para o problema de falta de segurança nas mensagens trafegadas pelo Direto. A solução se baseia na criação de uma infra-estrutura de chaves públicas. É apresentado todo o processo de criação e uso da infra-estrutura desde a avaliação da necessidade de sua criação, passando pela criação da Autoridade Certificadora e Autoridade Registradora, finalizando com o uso dos certificados pelos usuários do Direto.

5.1 Introdução

A proposta inicial deste trabalho era desenvolver uma infra-estrutura de chaves públicas completa, utilizando-se de alguma ferramenta já existente e de código fonte aberto. Com este propósito, foi feito um estudo sobre algumas destas ferramentas já implementados e que são distribuídas livremente.

As características esperadas destas ferramentas e que são a base para a proposta de solução dos problemas de segurança do Direto são:

- totalmente desenvolvida em código fonte aberto, para manter o padrão do Direto;
- facilmente acoplável ao software Direto;
- independente de plataforma, garantindo a característica de portabilidade do Direto;
- interface amigável;

As ferramentas mais próximas para a solução proposta foram as desenvolvidas pelos consórcios OpenCA e OpenSSL. Com essas ferramentas foi realizado um estudo mais aprofundado, explorando suas características. Os detalhes destas ferramentas estão descritas no capítulo 3, bem como a explicação de porque elas não foram utilizadas.

Como as ferramentas existentes não conseguiram alcançar todos os requisitos necessários à proposto desejada, a solução encontrada foi propor uma infra-estrutura de chaves públicas para o ambiente do Direto totalmente desenvolvida em Java, onde os serviços, características, funcionalidades e componentes fossem todos avaliados e projetados dentro da necessidade específica da Procergs.

5.2 Avaliando a necessidade de uma ICP

O Direto está sendo projetado [FAP2000] [PRO2000] com o objetivo principal de atender a demanda de um software de comunicação de baixo custo, que interligue os diversos órgãos do Estado. Além disso, o Direto poderá ser utilizado por quaisquer empresas e entidades que quiserem valer-se do conceito de Software Livre para implementar esta solução em seus estabelecimentos.

Nesse sentido, pode-se facilmente visualizar que muita informação sigilosa trafegará através de seu ambiente, sendo um alvo fácil para os *crackers*, que são usuários com um grande conhecimento sobre redes e comunicações, e que se utilizam do que sabem para explorar as fraquezas da segurança dos sistemas, com o objetivo de ler, copiar ou mesmo alterar informações importantes.

Na figura 5.1 visualiza-se a arquitetura do Direto, onde pode-se notar que, na fase atual de desenvolvimento, o projeto já faz uso de alguns protocolos para o tráfego das mensagens, porém, na quase totalidade, são protocolos inseguros.

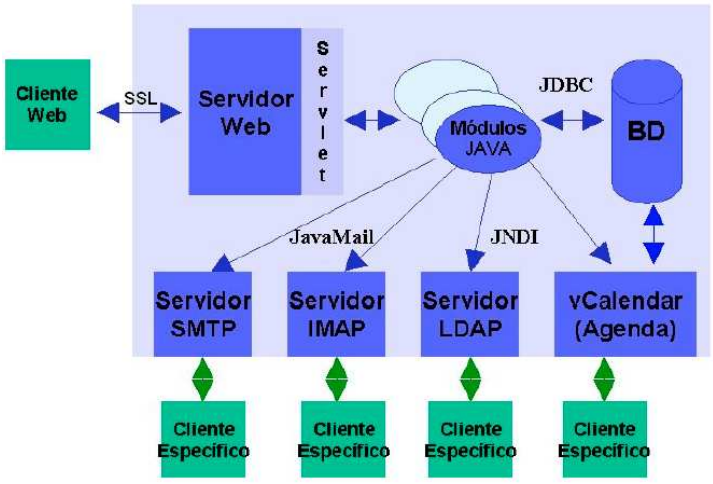

FIGURA 5.1 - Arquitetura do DiretoGNU

O protocolo utilizado para a transferência das mensagens é o SMTP, que é um protocolo inseguro [KLE2001], já que não utiliza nenhum tipo de verificação de autenticidade ou integridade da mensagem.

Conforme mostrado na figura 5.1, a única segurança se encontra no processo de tráfego de uma mensagem entre um Cliente Web e o Servidor Web, já que utiliza-se do protocolo SSL [FRE1996], que fornece criptografia de dados, autenticação de servidor e integridade de mensagem para transmissão de dados pela Internet

As mensagens trafegadas pelo Direto, ao chegarem ao servidor de e-mail, são armazenadas em texto puro, sem qualquer tipo de validação quanto à veracidade e integridade da informação. Qualquer pessoa mal-intencionada pode acessá-la no servidor de e-mail, lê-la ou mesmo alterá-la, sem que o receptor tenha qualquer desconfiança sobre seu sigilo ou a autenticidade de seu conteúdo.

Os usuários do Direto pertencem a vários níveis hierárquicos nas organizações onde ele já está instalado, de onde se conclui que o conteúdo das mensagens trafegadas varia muito, podendo ser desde mensagens sem necessidade alguma de segurança até mensagens com conteúdo sigiloso. Um intruso, ou espião, poderia enviar uma mensagem fazendo-se

passar por uma pessoa de um alto nível da hierarquia com a intenção de atrapalhar algum negócio em andamento ou mesmo com a intenção de obter informações sigilosas.

5.3 Definindo o modelo de confiança

O modelo de confiança [PER1999] é utilizado para descrever o relacionamento entre os usuários finais, partes verificadoras e a AC. Existem vários modelos de confiança definidos pela literatura e em uso por implementações de ICP.

O modelo proposto para a implementação da ICP da Procergs é um modelo híbrido de confiança, conforme mostra a figura 5.2 Um sistema híbrido permitirá à companhia utilizar dois modelos distintos de confiança, o modelo hierárquico e o modelo de certificação cruzada.

No modelo hierárquico, uma AC delega sua autoridade para uma ou mais autoridades subsidiárias. Como uma justificativa para este modelo está o fato de a Procergs poder delegar sua autoridade para outros órgãos ou empresas, como por exemplo para a Secretaria da Saúde, a SUSEPE, a METROPLAN e outros.

No modelo de certificação cruzada, uma AC ou grupo de ACs relacionadas podem operar com independência umas das outras. Estas ACs tem relacionamentos de confiança. Nesse modelo, cada AC determina qual o nível de relacionamento possui com cada uma das outras ACs. A justificativa para o uso desse modelo está na aceitação de certificados de alguma outra AC, fora do escopo da Procergs, como por exemplo, certificados criados pela SERPRO.

A figura 5.2 mostra um exemplo de modelo de certificação para a ICP da Procergs, onde pode-se visualizar algumas das empresas que já tem o produto instalado.

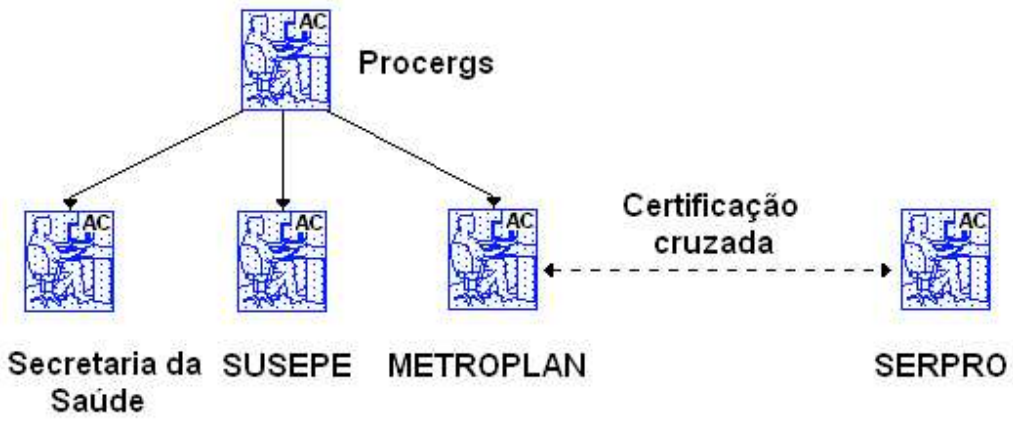

FIGURA 5.2 – Modelo de Confiança da ICP da PROCERGS

5.4Definindo a necessidade de ARs

Uma Autoridade Registradora (AR) pode ser considerada apenas como um componente estendido de uma ICP, porém, dependendo do crescimento das entidades envolvidas no processo de obtenção de certificados, sua presença se torna fundamental.

No modelo de ICP da Procergs, dois pontos foram fundamentais para a decisão de criar-se a entidade AR:

- a entidade AC deve ficar o mais isolada possível, já que é a parte de confiança fundamental em todo o processo. Com a implementação da entidade AR pode-se isolar a entidade AC, já que apenas a AR terá permissão de acesso ao servidor onde se localiza a AC;

- alguns dos vários órgãos do estado que possuem o Direto já instalado, possuem postos de trabalho espalhadas pelo Estado do Rio Grande do Sul. Cada órgão poderá definir algumas ARs, que poderão estar disponíveis dentro dos posto de trabalho, ficando responsáveis por verificar as informações de cada usuário que faz a solicitação de um certificado. Somente após atestada a validade das informações prestadas, a AR enviará a solicitação para a AC, evitando assim o gargalo na criação de certificados.

A figura 5.3 tem a intenção de mostrar a instalação de ARs em postos de trabalho, em algum dos órgãos que utilizam o Direto.

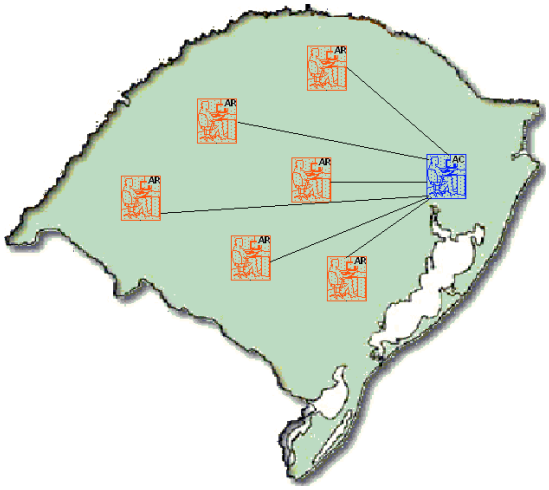

FIGURA 5.3 – Instalação das ARs

5.5 Certificados

Os certificados são o meio mais seguro de distribuir, para os usuários ou partes verificadoras (ACs ou ARs), as chaves públicas dentro de uma rede. Neste capítulo, serão propostos tanto o formato do certificado a ser utilizado pelo Direto, como os dados necessários para o correto funcionamento do Certificado.

5.5.1 Formato de Certificado

O formato de certificados mais amplamente aceito é o X.509v3 [BUR2002], por isto, este será o formato que será utilizado por todos os certificados criados e assinados pela AC da Procergs.

O sistema Direto já utiliza LDAP para armazenar todos os dados relativos aos usuários, seus contatos, assim como suas preferências sobre o sistema. O esquema utilizado pelo Direto precisa ser estendido para que possa armazenar as informações dos certificados e a chave pública dos usuários. A alteração do esquema e a criação dos novos tipos de dados são apresentadas abaixo:

5.5.1.1 Extensão do esquema

Para armazenar o certificado e a chave pública dos usuários, será necessária a extensão do esquema do LDAP do Direto, com a criação de dois novos atributos, diretoCertificado e usuarioPublicKey. Estes novos atributos farão parte do objeto diretoPerson, já existente no esquema atual, onde são armazenados todos os atributos do usuário. O objeto, já com os novos atributos, é apresentado a seguir.

```
objectclass ( 1.3.6.1.4.1.15509.1.2
      NAME  'diretoPerson'
      DESC  'objeto temporário para o direto. FIXME'
      SUP   top
      STRUCTURAL
            MUST (
                  objectClass $ cn $ uid $ acessodiscado $ empresa
                 )
            MAY (
                  jpegphoto $ userPassword $ userCertificate $ sn $ ou $
                  o $ l $ st $ c $ city $ matricula $ dataadmissao $
                  description $ setor $ cargo $ title $ telephoneNumber $
                  extension $ ramal $ facsimiletelephonenumber $ mail $
                  mailAcceptingGeneralID $ mailDrop $ givenname $
                  datademissao $ postalAddress $ postalCode $ homePhone $
                  bairro $ mobiletelephonenumber $ pagertelephonenumber $
                  placaveiculo $ datanascimento $ estadocivil $ sexo $
                  carteiraprofissional $ cpf $ identidade $
                  tituloeleitor $ pispasep $ reservista $ datafgts $
                  opcaofgts $ escolaridade $ naturalidade $
                  nacionalidade $ entradanopais $ filiacao $
                  nomeconjuge $ cpfconjuge $ profissaoconjuge $
```

```
                enderecoprofissionalconjuge $ nomefilho $
                datanascimentofilho $ horaacesso $ dataacesso $
                ultimoacesso $ apelido $ admempresa $
                diretoLastPasswords $ diretoLastPasswordChange &
                diretoCertificado $ diretoPublicKey
            )
        )
```

5.5.1.2 Criação dos atributos

Os dois atributos que foram estendidos no esquema já utilizado pelo Direto precisam ser definidos, conforme segue:

```
attributetype ( 1.3.6.1.4.1.15509.0.75
        NAME 'diretoCertificado'
        DESC 'usado para armazenar o certificado do usuário. FIXME'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.8
        )

attributetype ( 1.3.6.1.4.1.15509.0.76
        NAME 'diretoPublicKey'
        DESC 'usado para armazenar a chave pública do usuário. FIXME'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.6
        )
```

Os novos atributos criados são definidos pela RFC2252 [WAH1997] conforme a seguinte BNF (Backup-Naur-Form, sistema de escrita e expressão da sintaxe de uma linguagem de programação):

```
LdapSchema = "(" whsp
  numericoid       whsp
  [ "NAME"         qdescrs  ]
  [ "DESC"         qdescrs  ]
  [ "EQUALITY" woid     ]
  [ "SUBSTR"     woid     ]
  [ "SYNTAXES" OID      ]
  whsp ")"
```

- campo "numericoid" identifica o esquema unicamente. Um novo *numericoid* pode ser criado se qualquer um dos campos do esquema mudar. Esse campo pode ser comparado a um número seqüencial para cada campo do esquema;

- campo "NAME" contem um *label* opcional para o esquema, que pode ser interpretado por um usuário;

- campo "DESC" contém uma descrição opcional para o esquema, que pode ser interpretada por um usuário;

- campo "EQUALITY" é utilizado por servidores para comparar atributos contra valores inseridos quando realizadas pesquisas ou operações de comparação;

- campo "SUBSTR" é utilizado por servidores para comparar atributos contra valores inseridos quando são realizadas pesquisas ou operações de comparação;

- campo "SYNTAX" lista a sintaxe dos OIDs definidos neste esquema. A RFC2252 [WAH1997] possui uma lista de OIDs pré-definidos, que podem ser utilizados no esquema.

Para a definição da sintaxe foram utilizadas algumas regras de codificações especiais, conforme a definição a seguir:

- whsp = [SPACE]

- qdescrs = qdescr / (whsp "(" qdescrlist ")" whsp)

- qdescr = whsp """ descr """ whsp

- qdescrlist = [qdescr *(qdescr)]

- descr = keystring

- keystring = a [anhstring]

- anhstring = 1*k

- k = a / d / "-" / ";"

- woid = whsp oid whsp

- oid = descr / numericoid

- numericoid = numericstring *("." numericstring)

- numericstring = 1*d

- d = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"

- a = "a" / "b" / "c" / "d" / "e" / "f" / "g" / "h" / "i" / "j" / "k" /
  "l" / "m" / "n" / "o" / "p" / "q" / "r" / "s" / "t" / "u" / "v" /
  "w" / "x" / "y" / "z" / "A" / "B" / "C" / "D" / "E" / "F" /
  "G" / "H" / "I" / "J" / "K" / "L" / "M" / "N" / "O" / "P" /
  "Q" / "R" / "S" / "T" / "U" / "V" / "W" / "X" / "Y" / "Z"

5.5.2 Estrutura dos Certificados

Nesta seção estão descritos todos os campos de um certificado X.509v3 assim como as extensões e os valores que serão gravados em cada um destes campos para o projeto Direto. A tabela 5.1 faz a definição de cada um dos campos de um certificado e quais os valores utilizados para o Direto.

TABELA 5.1 – Definição dos campos do certificado

Campos			Definição	Valor p/ Direto
tbsCertList	version		diferencia os certificados pela versão em que foram criados	2 - representa o certificado X.509v3
	SerialNumber		identificador único para cada certificado emitido pela AC	contém um número seqüencial, criado no momento em que é gerado o certificado
	signature	algorithm	identifica o algoritmo utilizado para assinar o certificado	gravada quando da configuração da AC
		parameters	parametros adicionais utilizados pelo algoritmo	não utilizado para o Direto
	issuer		identifica o nome distinto (DN) com a qual a AC cria e assina o certificado;	DN da AC-Pro
	validity	notBefore	intervalo de tempo onde o certificado é valido	será a data da criação do certificado
		notAfter		será o prazo final calculado a partir da data inicial, considerando o prazo de validade do certificado
	subject		identifica o DN da entidade final que mantém a chave privada correspondente	DN do usuário
	subjectPublicKeyInfo	algorithm	identificador de algoritmo	
		subjectPublicKey	contém o valor da chave pública do sujeito	
	issuerUniqueId		campo opcional	não será utilizado pela AC-Pro
	extensions		extensões adicionais para o tratamento do certificado	conforme definido a seguir
signatureAlgorithm			o mesmo que gravado em signature	o mesmo que gravado em signature
signatureValue			assinatura do certificado	assinatura do certificado

Com a crescente utilização de certificados, algumas características necessárias foram sendo incorporadas ao formato inicial do certificado, chamados de *extensions* (extensões), de forma a acrescentar segurança aos seus usuários e ampliar a interoperabilidade entre componentes de uma ICP e entre ICPs. As informações contidas nos campos de extensão podem ser marcadas como críticas ou não. Algumas da extensões disponíveis para certificados são detalhadas a seguir, indicando se haverá uso pelo Direto ou não:

- Authority Key Identifier: utilizada [HOU1999] como um meio de identificar a chave pública correspondente à chave privada usada para assinar o certificado. Esta extensão é utilizada quando o assinante tem múltiplas chaves de assinatura. A RFC2459 recomenda o uso desta extensão, porém ela não precisa ser marcada como crítica. A especificação da extensão é definida como:

  ```
  AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier [0] KeyIdentifier OPTIONAL,
    authorityCertIssuer [1] GeneralNames OPTIONAL,
    authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL}
  ```

  Para o Direto, os valores serão definidos como:

  ```
  AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier: contém o resumo SHA-1 da chave pública da AC-Pro,
    authorityCertIssuer: DN da AC-Pro,
    authorityCertSerialNumber: número sequencial do certificado}
  ```

- Subject Key Identifier: Quando uma entidade final obtém vários certificados, podendo ser eles de múltiplas ACs, esta extensão provê o meio de rapidamente identificar o conjunto de certificados que contém uma determinada chave pública. É útil quando o titular do certificado é uma certificadora (o certificado inclui a extensão *basic consraints* com o valor CA=TRUE). A RFC2459 ordena o uso desta extensão. A especificação da extensão é definida como:

  ```
  SubjectKeyIdentifier ::= KeyIdentifier
  ```

  Para o Direto, o valor será definido como:

  `SubjectKeyIdentifier`: resumo SHA-1 da chave pública da AC-PRO

- Key Usage: esta extensão define a proposta (cifragem, assinatura...) para a qual a chave pública do certificado pode ser utilizada. Os valores válidos para a proposta de uso da chave são apresentados na tabela 5.2:

TABELA 5.2 – Valores válidos para *Key Usage*

Proposta	Valor	Definição
digitalSignature	0	assinatura digital
nonRepudiation	1	assinatura digital com propósito de não-repúdio
keyEncipherment	2	cifragem de chaves
dataEncipherment	3	cifragem
keyAgreement	4	acordo de chaves
keyCertSign	5	usado em certificados de ACs, serve para verificar a assinatura do certificado
cRLSign	6	quando a chave é utilizada para verificar a assinatura de LCRs
encipherOnly	7	utilizada para os propósitos de cifragem de dados em conjunto com acordo de chaves
decipherOnly	8	utilizado com o propósito de decifragem de dados em conjunto com acordo de chaves

A especificação da extensão é definida como:

```
KeyUsage ::= BIT STRING {
          digitalSignature (0),
          nonRepudiation (1),
          keyEncipherment (2),
          dataEncipherment (3),
          keyAgreement (4),
          keyCertSign (5),
          cRLSign (6),
          encipherOnly (7),
          decipherOnly (8) }
```

Para o Direto, o valor será definido como:

```
KeyUsage ::= BIT STRING {
         digitalSignature (0),
          nonRepudiation (1),
          keyEncipherment (2),
          dataEncipherment (3) }
```

- Extended Key Usage: Esta extensão indica uma ou mais propostas para a qual a chave pública do certificado pode ser utilizada. Se esta extensão estiver presente, então o certificado deve apenas ser usado para uma das propostas indicadas. Um certificado pode possuir esta extensão em conjunto com a extensão Key Usage ou no lugar da extensão Key Usage. Se um certificado contiver tanto a Key Usage como a Extended Key Usage, elas devem ser processadas independentemente e os certificados devem apenas ser utilizados após ambas serem validadas.

  A especificação da extensão é definida como:

```
ExtKeyUsageSyntax ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
KeyPurposeId ::= OBJECT IDENTIFIER
```

  Para o Direto, o valor será definido como:

KeyPurposeId : *E-mail seguro*

- CRL Distribution Point: esta extensão identifica como a informação da CRL é obtida. Esta extensão, por sua vez, é definida como uma seqüência de DistributionPoint, que consiste de três campos, cada um deles opcional;

  - DistributionPoint: quando este campo está presente, ele contém uma seqüência de um ou mais nomes, como uma URI ou uma NameRelativeToCRLIssuer;

  - reasons: indica o motivo da revogação do certificado. Os valores permitidos para este campo são apresentados na tabela 5.3.

TABELA 5.3 – Valores válidos de ReasonCode

Razão	Valor	Propósito
unused	0	certificado não usado
keyCompromise	1	comprometimento de chave
cACompromise	2	compometimento da chave da AC
affiliationChanged	3	troca de nome
superseded	4	substituído
CessationOfOperation	5	término de operação da AC
certificateHold	6	certificado suspenso

  - cRLIssuer: este campo é apresentado apenas quando o emissor do certificado não é o mesmo emissor da CRL. Neste campo será apresentado o emissor da CRL. Se os emissores forem os mesmos, então este campo é omitido e o campo DistributionPoint deve ser apresentado.

A especificação da extensão é definida como:

```
DistributionPoint ::= SEQUENCE {
distributionPoint [0] DistributionPointName OPTIONAL,
reasons [1] ReasonFlags OPTIONAL,
cRLIssuer [2] GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
fullName [0] GeneralNames,
nameRelativeToCRLIssuer [1] RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
unused (0),
keyCompromise (1),
cACompromise (2),
affiliationChanged (3),
superseded (4),
cessationOfOperation (5),
certificateHold (6),
privilegeWithdrawn (7),
aACompromise (8) }
```

Para o Direto, o valor será definido como:

```
DistributionPointName:
URL=https://www.direto.org/documentos//crl/crl_basica.crl
ReasonFlags:= keyCompromise (1)
```

- Private Key Usage Period: esta extensão não precisa ser utilizada dentro da ICP-Pro. Semelhante ao campo Validity, essa extensão indica o período de tempo para a utilização da chave privada associada à chave pública nesse certificado. Essa extensão não será utilizada;

- Certificate Policies: contém uma seqüência de um ou mais itens de identificação de políticas. Cada item consiste de um identificador de objeto (OID) e, opcionalmente, qualificadores que especializam a política correspondente. Em um certificado de entidade final, a extensão descreve o conjunto de políticas sob as quais o certificado foi emitido, incluindo o seu uso pretendido. Usuários de certificados devem conhecer as políticas aceitáveis para suas aplicações. Em um certificado de entidade certificadora, a extensão descreve as políticas que podem ser incluídas entre as políticas vigentes para certificados que lhe sejam subordinados numa cadeia de certificação.

A RFC2459 define dois tipos de qualificadores de política para o uso dos redatores das políticas de certificados e emissores de certificados. Os tipos de qualificadores são:

- CPS Pointer: o ponteiro CPS contém um ponteiro para uma CPS publicada pela AC da Procergs. O ponteiro tem a forma de uma URI;

- User Notice tem a finalidade de permitir à aplicação exibir ao usuário texto informativo sobre a política de certificação do emissor do certificado.

  A especificação da extensão é definida como:

  ```
  PolicyQualifierId ::= OBJECT IDENTIFIER ( id-qt-cps)

  Qualifier ::= CHOICE { cPSuri CPSuri }

  CPSuri ::= IA5String
  ```

  Para o Direto, o valor será definido como:

  ```
  CPSuri: https:\\www.direto.org\documentos\cps_pro
  ```

- Policy Mappings: Essa extensão é usada em certificados de entidades certificadoras. Constitui-se de uma lista de pares de OIDs (identificadores únicos) correspondentes as políticas. Esta extensão não será utilizada pelo Direto;

- Subject Alternative Name: essa extensão permite uma ou mais formas alternativas de nomes associados ao proprietário do certificado. A utilização deste campo permite o suporte dentro de vários aplicativos que empreguem formas próprias de

nomes, como endereço de e-mail, DNS, endereço IP ou URI.. Os tipos de nomes alternativos possíveis são apresentados na tabela 5.4.

TABELA 5.4 - Tipos possíveis de Nomes Alternativos

Tipo de Nome	Valor
OtherName	0
Rfc822Name	1
DNSName	2
X400Address	3
DirectoryName	4
EdiPartyName	5
UniformResourceIdentifier	6
IPAdress	7
RegisteredID	8

A especificação da extensão é definida como:

```
SubjectAltName ::= GeneralNames

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE
{ otherName [0] OtherName,
     rfc822Name [1] IA5String,
     dNSName [2] IA5String,
     x400Address [3] ORAddress,
     directoryName [4] Name,
     ediPartyName [5] EDIPartyName,
     uniformResourceIdentifier [6] IA5String,
     iPAddress [7] OCTET STRING,
     registeredID [8] OBJECT IDENTIFIER }
```

Para o Direto, o valor será definido como:

```
SubjectAltName: rfc822Name, que é o endereço de e-mail do usuário
```

- Issuer Alternative Name: extensão que indica uma ou mais formas alternativas de nome associado ao emissor deste certificado. Funciona da mesma maneira que o Subject Alternative Name. Essa extensão não será utilizada pelo Direto;

- Subject Directory Attributes: essa extensão é utilizada para transportar atributos de identificação. Este campo deve ser marcado como não crítico. Essa extensão não será utilizada pelo Direto;

- Basic Constraints: essa extensão indica se o proprietário do certificado é uma AC, podendo também especificar um comprimento para o caminho de certificação. A RFC2459 ordena que essa extensão esteja presente e marcada como crítica em todos

os certificados de AC. Esta extensão será utilizada pelo Direto apenas para o certificado raiz;

- Name Constraints: Esta extensão, que deve ser usada apenas em certificados AC, indica um espaço de nomes dentro da qual todos os certificados subsequentes de um caminho de certificação devem se localizar. Esta extensão não será utilizada pelo Direto;

- Policy Constraints: Esta extensão pode ser usada em certificados emitidos para ACs. Esta extensão restringe o caminho de validação em duas formas. Ele pode ser usado para proibir mapeamento de políticas ou obrigar que cada certificado em um caminho contenha um identificador de política. Esta extensão não será utilizada pelo Direto.

5.6 Definindo a ICP da Procergs

A Infra-estrutura de chaves públicas da Procergs, ICP-Pro, será mais que um servidor de certificados, pois será responsável pela administração de todo o ciclo de vida dos certificados.

A figura 5.4 a estrutura da ICP-Pro, com seus componentes principais.

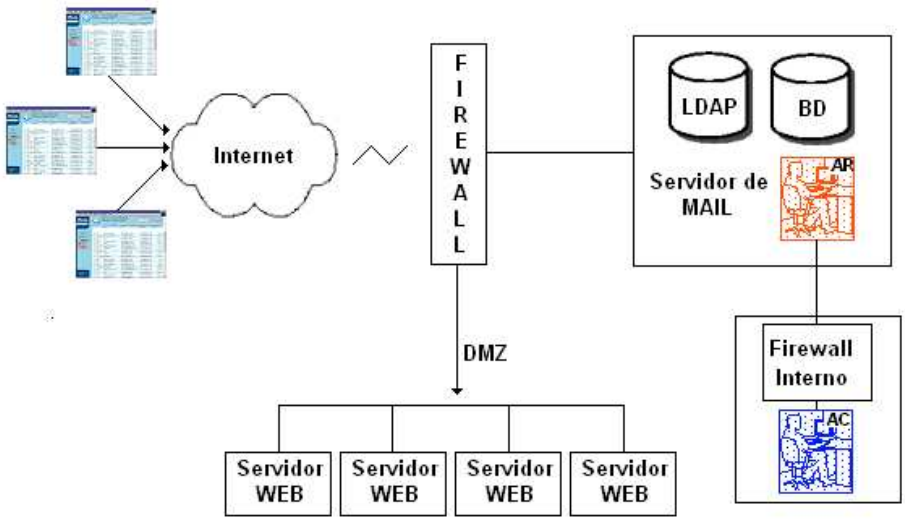

FIGURA 5.4 – Principais componentes da ICP-Pro

A Autoridade Certificadora (AC) estará instalada em um servidor, protegido por um Firewall Interno. A única entidade que terá acesso a este servidor é a Autoridade Registradora (AR). Várias ARs podem ser instaladas.

A seguir serão detalhadas as funcionalidades disponíveis pela ICP-Pro, cada uma das entidades da ICP-Pro e as tarefas executadas por cada uma delas.

5.6.1 Funcionalidades da ICP-Pro

As funcionalidades que estarão disponíveis após a implantação da ICP-Pro são descritas nas seções à seguir.

5.6.1.1 Geração do Par de Chaves em favor do usuário

Existem duas formas distintas de criação do par de chaves do usuário. Uma das formas é solicitar que o próprio usuário crie seu par de chaves. Quando o usuário faz a solicitação de geração de um certificado, ele envia, junto com seus dados, a sua chave pública. Nesse método, a garantia de não-repúdio é mais eficaz, já que a chave privada é gerada pelo usuário e nunca sai de seu poder. Porém, como o usuário se conecta ao servidor via um navegador, independente de lugar ou máquina, ele é obrigado a carregar sua chave privada para poder utilizá-la.

Para este projeto, então, optou-se por utilizar um segundo método conhecido, que é a própria AC gerar o par de chaves e armazená-las, garantindo a sua segurança.

Quando o usuário requisita a geração de um certificado, o sistema verifica se o usuário já possui um par de chaves geradas para ele. Se não possuir, o sistema gera este par de chaves utilizando as configurações de método de criptografia e tamanho de chaves, registrado no módulo de configuração da AC-Pro.

5.6.1.2 Armazenamento da chave pública

A chave pública do usuário será armazenada no diretório LDAP, no atributo diretoPublicKey. Como a chave pública pode ser distribuída livremente, ela pode ser armazenada sem qualquer tipo de método adicional de segurança.

5.6.1.3 Armazenamento da chave privada

O ponto principal dos sistemas criptográficos de chave pública e seus certificados conta significativamente com a segurança das chaves privadas. É fundamental que apenas o proprietário da chave privada possa utilizá-la. Para que isso seja garantido, o armazenamento da chave privada será no banco de dados PostgreSQL. Quando o usuário necessitar utilizar esta chave, ela será entregue a ele através de um canal seguro entre o usuário e o banco de dados.

Simplesmente armazenar a chave privada no banco de dados não é seguro, então, será utilizado uma criptografia baseada em senha para armazená-la. A criptografia baseada em senha, ou PBE – *Password-Based Encryption* [GAR2001] utiliza uma combinação de *hash* e criptografia simétrica. É criado um resumo da senha utilizando-se um algoritmo de

hash como SHA-1 e então este resultado é usado para construir uma chave binária para um outro algoritmo, como o IDEA.

Um dos problemas com a criptografia baseada em senhas é a possibilidade de criar uma lista pré-compilada de senhas, criar um *hash* destas senhas e então ter um conjunto de chaves prontas para aplicar nos dados cifrados. Isto permitiria a um espião aplicar todas as chaves normalmente utilizadas e então, muito rapidamente, determinar se alguma delas decifra os dados. Existe uma técnica usada para combater este ataque conhecida como *salting* [GAR2001]. Um *salt* é um valor randômico agregado à senha antes de criar o *hash* para criar a chave.

O *salt* é armazenado com os dados que são cifrados. Para a decifragem, o *salt* deve ser extraído dos dados cifrados e então combinado com uma senha para criar a chave de decriptografia.

A criptografia requer uma senha e o texto puro a ser cifrado (no caso, a chave privada do usuário). Então, é criado um *salt* que é armazenado concatenado ao texto cifrado. A figura 5.5 mostra o processo de criptografia.

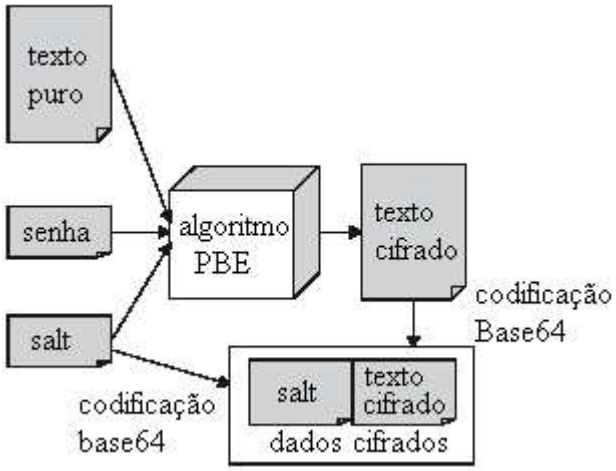

FIGURA 5.5 – Criptografia baseada em senha

Para a decifragem é necessário pegar o bloco de dados cifrados e separar ele em *salt* e texto cifrado. Então, a senha e o *salt* podem ser usados para inicializar o algoritmo que decifra os dados, gerando novamente o texto puro (a chave privada do usuário). A figura 5.6 mostra o processo de decriptografia.

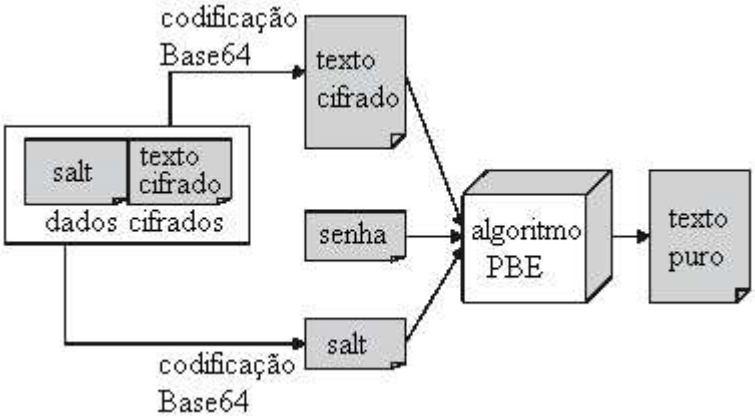

FIGURA 5.6 – Decriptografia baseada em senha

Uma boa senha a ser utilizada poderia ser a própria senha do usuário no Direto. Porém, usuários costumam usar senhas fáceis de serem lembradas, e por isso, fáceis de serem quebradas. Então, uma boa senha seria mesclar alguns dados pessoais do usuário, como a senha, a carteira de identidade e seu CPF.

5.6.1.4 Definição do canal seguro para transferência da chave

O protocolo mais utilizado [KAR1996] para segurança no canal de comunicação entre duas aplicações que conversam sobre a Internet é o SSL (Secure Socket Layer). O padrão SSL possui três funcionalidades importantes para a segurança:

- provê autenticação, que ajuda a garantir a legitimidade das entidades envolvidas em um diálogo;

- provê privacidade, que ajuda a garantir que um terceiro não pode decifrar o diálogo entre duas entidades;

- mantém integridade, utilizando um MAC (Message Authentication Code), que é similar a um *checksum*, que ajuda a garantir que o diálogo entre as duas entidades não é modificado por um terceiro.

SSL usa criptografia de chave privada para cifrar os dados trocados entre as aplicações, permitindo um canal seguro e cifrado, chamado de túnel de dados.

A definição do túnel de dados utilizado para a transferência da chave privada do usuário, entre o banco de dados PostgreSQL e o cliente, será implementado através do pacote JSSE (Java Secure Socket Extensions) [SUN2003], que implementa comunicação segura na Internet através do protocolo SSL.

5.6.1.5 Ciclo de vida do par de chaves

Um par de chaves possui limitações quando a seu tempo de validade. Algumas destas limitações foram apresentadas na seção 2.4.5.

Uma chave privada que é utilizada [BUR2002] com o propósito de prover não-repúdio requer um armazenamento seguro por toda a vida útil da chave. Se esta chave for perdida, um novo par de chaves deve ser gerado. Depois que a chave expirar, ela deve ser seguramente destruída. A chave pública correspondente deve ser armazenada para poder autenticar os dados assinados com essas chaves privadas. Uma chave privada utilizada para fornecer criptografia deve ser arquivada para mais tarde fornecer a decriptografia dos dados legados cifrados.

No modelo inicial proposto para o Direto, estaremos levando em conta que o mesmo par de chaves será utilizado para a criptografia e para a assinatura. Nesse caso, tanto a chave privada como a chave pública devem ser armazenadas para posterior utilização. Uma nova proposta de criação de certificados deverá levar em conta a utilização de certificados diferenciados com pares de chaves próprios para suporte à criptografia e assinatura.

O tempo de validade de uma chave será o mesmo do certificado, a menos que a chave seja perdida ou se torne inválida por tentativa ou interceptação, quando então ela será imediatamente revogada juntamente com o certificado.

O mecanismo de armazenamento das chaves expiradas não será tratado nesse modelo.

5.6.1.6 Criação de Certificados

Para a criação dos certificados pela ICP-Pro, este projeto levará em conta que todos certificados a serem criados serão para uso pelo software Direto, portanto, os solicitantes serão usuários já cadastrados para uso do sistema, não sendo necessária a criação de uma interface de solicitação de criação de certificados específica, que possa ser acessada via navegador por qualquer usuário interessado em obter um certificado da AC-Pro. Todas as solicitações serão feitas através da interface do Direto, que entregará a solicitação para a Autoridade Registradora fazer a devida verificação.

O método de recepção da Requisição dos Certificados da Autoridade Registradora é detalhado no item 5.5.2, que trata da Autoridade Registradora.

A tarefa de criar o certificado consistirá de receber a requisição, que já vem com os dados do usuário validados pela Autoridade Registradora e com a chave publica do usuário, tendo apenas que assinar o certificado, armazená-lo e então avisar a Autoridade Registradora que o certificado foi devidamente gerado.

As figuras 5.7 e 5.8 mostram o Estudo de Caso da Criação de Certificados.

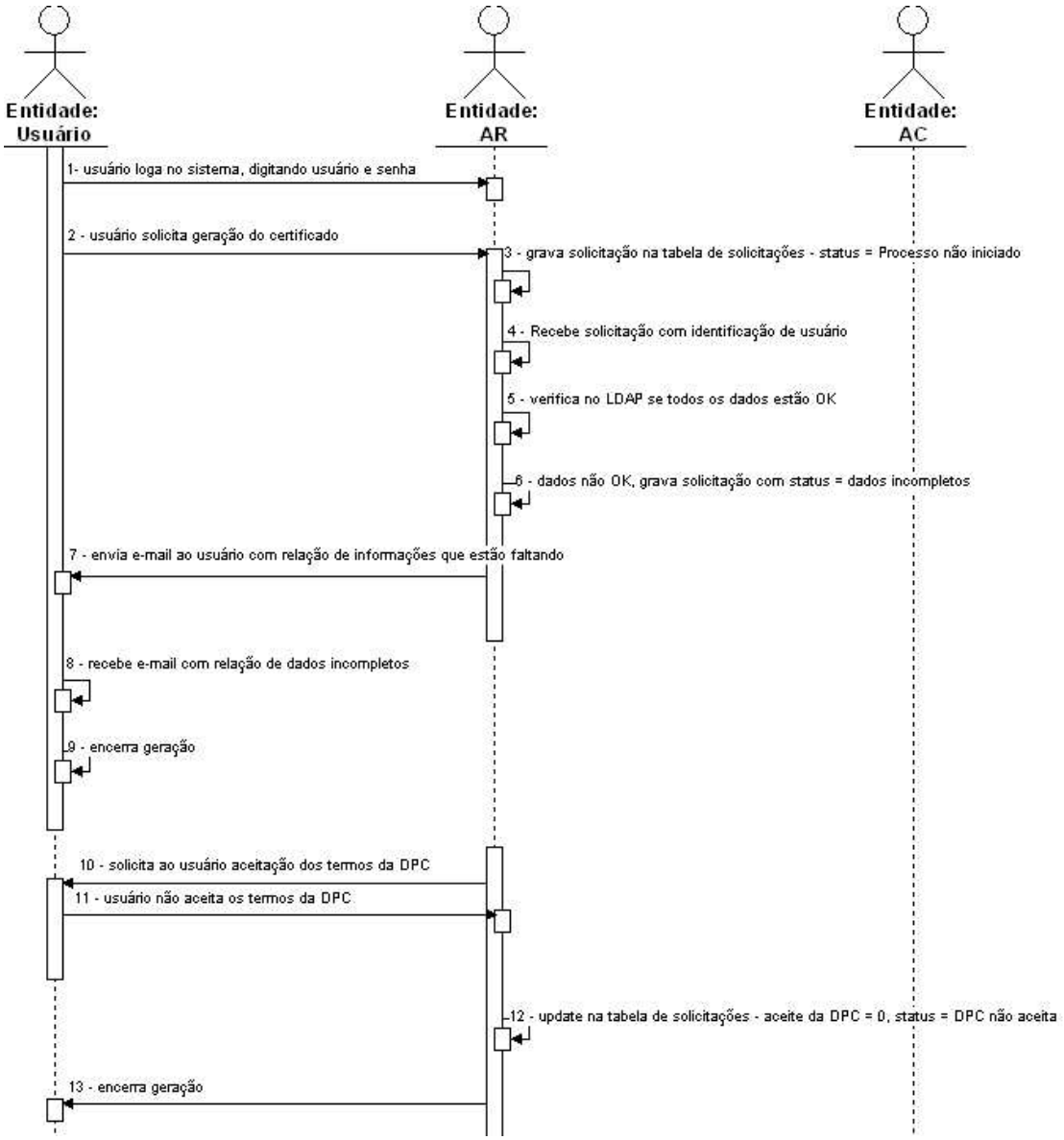

FIGURA 5.7 - Estudo de Caso – Geração de Certificados

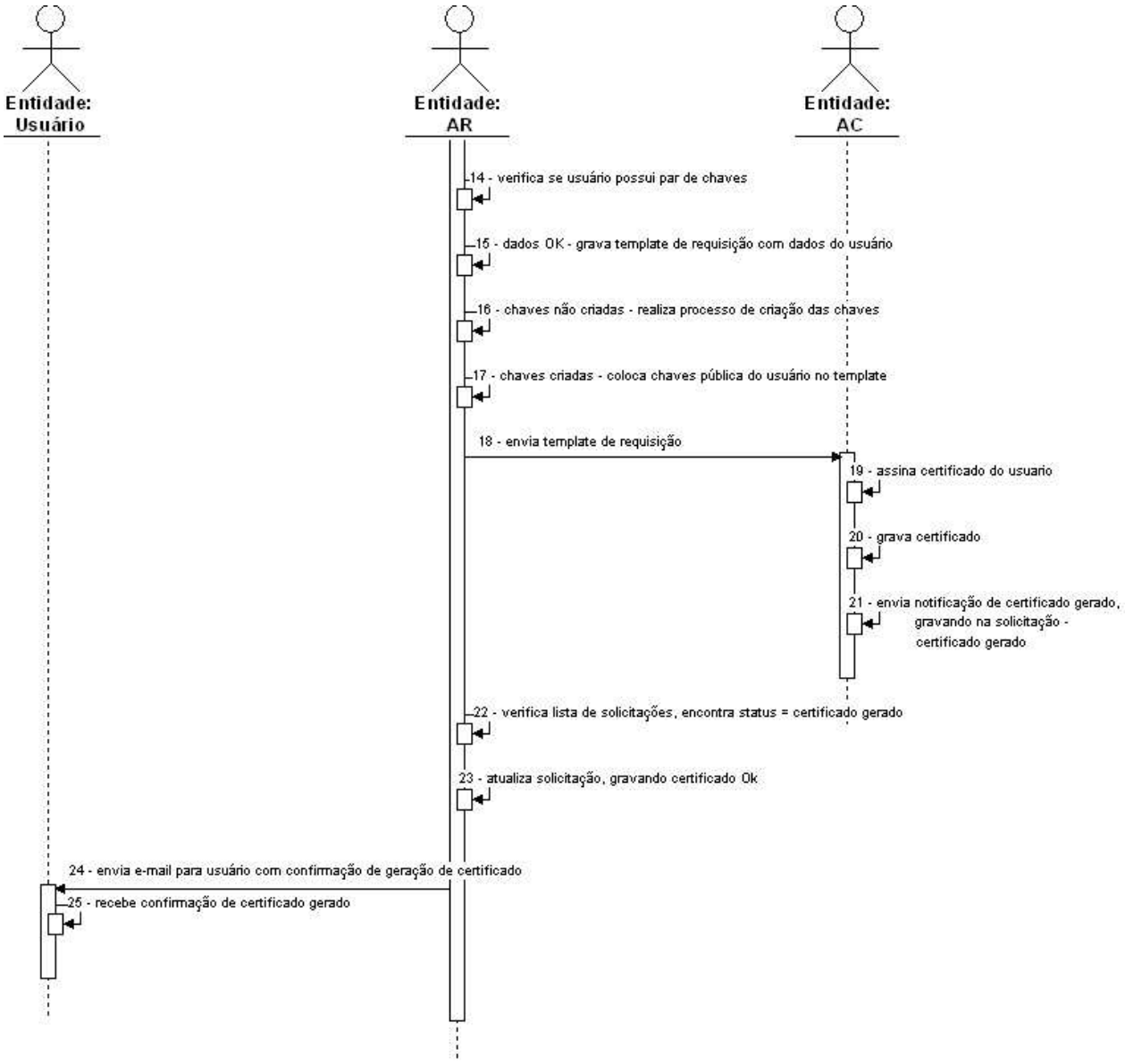

FIGURA 5.8 – Continuação do Estudo de Caso – Geração de Certificados

5.6.1.7 Armazenamento de Certificados

O armazenamento do certificado será no diretório LDAP, na informação referente ao sujeito do certificado. O esquema do Direto já foi alterado, visando a inclusão do atributo diretoCertificado e diretoPublicKey, conforme detalhado na seção 5.4.1.1.

5.6.1.8 Exportação de Certificado

A opção de exportação de um certificado tem a finalidade de permitir que um certificado seja enviado a outra AC, para obter sua assinatura. Se, por exemplo, a Procergs

optar por ser uma Autoridade Certificadora vinculada a ICP-Brasil, ela deverá exportar seu certificado e enviá-lo a ICP-Brasil para a sua assinatura.

5.6.1.9 Importação de Certificado

A opção de importação de um certificado tem a finalidade de permitir que um certificado seja importado após ele ter sido assinado por uma outra AC (por exemplo, pela ICP-Brasil).

5.6.2 Autoridade Certificadora AC-Pro

A Autoridade Certificadora da Procergs passa a ter, à partir de agora, o nome de AC-Pro e será a responsável por assinar e criar os certificados. Ela é, também, a responsável por criar e alterar as Listas de Certificados Revogados (LCR). A AC-Pro pode responder a requisições feitas diretamente a ela, ou responder requisições feitas pela Autoridade Registradora.

Para seu completo funcionamento, várias são as atividades que ela deve executar, que serão detalhadas a seguir:

5.6.2.1 Configuração da AC-Pro

Antes da AC-Pro começar a atender requisições, ela precisa ser configurada. Para isso, a AC-Pro precisa gerar seu próprio par de chaves (pública e privada). Para a criação dessas chaves será utilizado o algoritmo RSA com tamanho de chave de 2048 bits. A escolha do algoritmo RSA se deu pelo fato de ser um algoritmo bem conhecido, sendo instalado pelos navegadores mais utilizados e que, pelo tamanho da chave escolhida, é muito seguro.

A chave pública e a chave privada serão armazenadas em arquivos texto, no próprio servidor da AC-Pro. A chave pública estará armazenada no arquivo AC_PUB.KEY e a chave privada no arquivo AC_PRIV.Key. A chave pública será armazenada também no diretório público do Direto, conforme definido na seção 5.6.1.4.

5.6.2.2 Criação do Certificado Auto-Assinado

Certificados auto-assinados ou certificados *Root* são um tipo especial de certificados, já que não existe uma AC que assine o certificado. Neste caso, o Emissor e o Sujeito do certificado são os mesmos.

Um certificado auto-assinado é válido e seguro, mas os usuários preferem ter um certificado assinado por uma AC conhecida. A Procergs pode optar em criar um certificado auto-assinado ou solicitar, através de uma requisição de certificados, que a ICP Brasil assine seu certificado raiz. Vale salientar que um certificado auto-assinado pode não obter as mesmas funcionalidades de um certificado assinado pela ICP-Brasil. Um certificado

auto-assinado pode não ser automaticamente reconhecido pelos navegadores dos usuários, como por exemplo, os certificados da Verisign.

Para esse trabalho, o método selecionado foi o de criar o certificado auto-assinado, que serve a todos os propósitos da criação da ICP-Pro e possui custo zero.

O certificado criado será armazenado em um arquivo texto, chamado de AC_PRO.cert, que estará localizado no próprio servidor da AC-Pro.

5.6.2.3 Atendimento de Requisições da AR-Pro

A AC-Pro é solicitada a criar e assinar um certificado através da Autoridade Registradora. Se um usuário fizer a solicitação do seu certificado, esta solicitação será encaminhada à Autoridade Registradora. A Autoridade Registradora então envia os dados do usuário para a AC-Pro no formato de um *template,* chamado de Requisição de Certificados.

5.6.2.4 Validação do Certificado

O processo de validação de um certificado vai além da verificação de seu período de validade. Um certificado precisa ser verificado perante um Lista de Certificados Revogados (LCR).

Se, temporariamente, a LCR estiver indisponível e a informação de revogação não for obtida, este certificado não poderá ser validado. Neste caso, o sistema deverá apresentar ao usuário uma mensagem, esclarecendo o problema. A aceitação ou não do certificado como sendo confiável deverá ficar a cargo do usuário.

5.6.2.5 Renovação de Certificados

A renovação de um certificado pode ocorrer com ou sem a intervenção do usuário. Na configuração da AC-Pro dois parâmetros necessitarão ser preenchidos para o controle da renovação dos certificados. O primeiro parâmetro indica se a renovação do certificado se dará de forma automática. O segundo parâmetro, que só será preenchido no caso da renovação do certificado ficar por conta do usuário, trata do prazo em que o usuário será avisado sobre a expiração do certificado.

A AC-Pro será responsável por verificar os prazos de validade dos certificados.

Se for configurado que o certificado será renovado automaticamente, então o sistema revoga o certificado atual e cria novo certificado, com as mesmas informações e a mesma chave do certificado anterior. O usuário é apenas avisado, através de e-mail, que ocorreu a renovação automática de seu certificado.

Se a intervenção do usuário for solicitada, o sistema envia um e-mail ao usuário avisando sobre o prazo final de validade do seu certificado e convida o mesmo a realizar a sua renovação.

A solicitação de renovação é enviada para a Autoridade Registradora, que a envia a AC-Pro.

5.6.2.6 Revogação de Certificados

Um certificado é criado para estar confiável durante todo o seu período de validade. Porém, em alguns casos, um certificado ainda em vigor não deverá mais ser utilizado. Alguns desses fatores podem ser:

- anulação por troca de nome: como os usuários do Direto são pessoas físicas, esta troca de nome poderá ocorrer quando do casamento de uma funcionária, o que acarreta a troca de sobrenome;

- anulação por troca de empresa: no caso de um funcionário ser locado para trabalhar em outro órgão, que não aquele a que seu certificado foi criado, o sistema deverá ser solicitado a revogar o certificado em uso;

- comprometimento da chave privada: na possibilidade de que a chave privada de um usuário possa ter sido violada, o sistema deverá ser solicitado a revogar o certificado atual. Nesse caso, o par de chaves do usuário deverá ser recriado. A chave privada do usuário deverá ser armazenada com a intenção de decifrar documentos antigos que tenham sido cifrados com a respectiva chave pública, antes do seu comprometimento. E a chave pública deverá ser armazenada para que mensagens assinadas utilizando aquela chave possam ser lidas. No caso de a chave privada da própria AC-Pro ter sido violada, todos os certificados de usuário deverão ser revogados e os usuários deverão ser avisados. A AC-Pro então, deverá reiniciar seu processo, desde sua inicialização. Os certificados dos usuários deverão, todos, serem recriados;

- O dono do certificado não quer mais utilizá-lo: a solicitação de revogação virá do próprio usuário. Neste caso, o certificado será apenas revogado;

Em todos estes casos, o mecanismo é o mesmo. O usuário seleciona a opção de revogar certificado e entra com sua identificação. Uma lista dos certificados disponíveis para aquele usuário é apresentada para que ele possa selecionar o certificado que deseja revogar, informando também o motivo da revogação. Esta solicitação é enviada para a Autoridade Registradora, que valida as informações. Se tudo estiver correto, a solicitação é enviada à AC-Pro, que revoga o certificado. Após a revogação, o certificado é removido do diretório LDAP. A lista de todas as solicitações de revogações são mantidas na tabela de solicitações, que conterá o motivo da revogação, o nome e a função da pessoa que solicitou a revogação.

A tarefa de anular o certificado constitui-se de colocar o certificado (mais precisamente o número de série do certificado junto com outras informações) em uma Lista de Certificados Revogados (LCR).

A figura 5.9 mostra o Estudo de Caso da Revogação de Certificados.

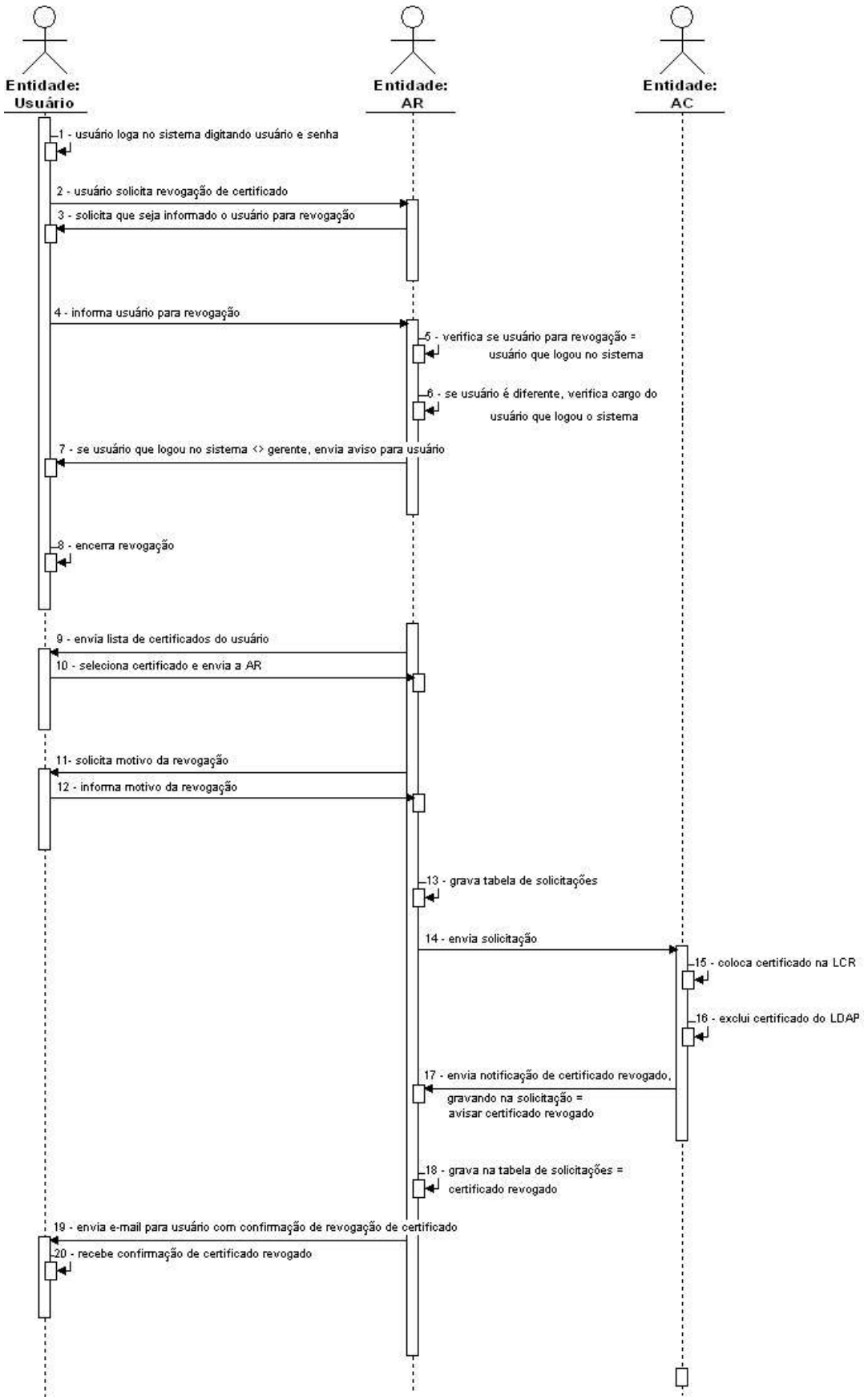

FIGURA 5.9 – Estudo de Caso da Revogação de Certificados

5.6.2.7 Manutenção da LCR

A AC necessita de uma forma segura de revogar um certificado e manter os usuários avisados sobre esta revogação. A forma mais conhecida [BUR2002] para tratar revogações de certificados é conhecida como LCR. Uma LCR é uma lista gerada pela AC que contém informações únicas sobre os certificados revogados e que permite às partes confiáveis determinar se um certificado é válido ou não.

A LCR deve ser disponibilizada em um repositório disponível publicamente.

O tempo de publicação da LCR é crítica, principalmente durante o período de latência entra a AC revogar um certificado e suas subsequentes publicações, onde o usuário pode acabar aceitando um certificado como válido, embora já tenha sido revogado.

Cada LCR tem um escopo definido. O escopo de uma LCR, por exemplo, pode ser “todos os certificados emitidos pela AC-Pro”.

O esquema de revogação de certificados [LIA2002] com o uso de uma LCR pode ser definido em alguns pequenos passos. No primeiro passo, a lista dos certificados revogados é gerada periodicamente pela AC. Cada lista tem dois campos de datas chamados de *thisUpdate* e *nextUpdate*. Uma LCR é dita válida se a data atual é maior ou igual a thisUpdate e menor ou igual a nextUpdate. Uma LCR que não está neste intervalo não pode ser utilizada para validar os certificados.

No segundo passo, a AC pode enviar uma LCR gerada para um repositório onde todas as entidades finais tem acesso. Nos certificados do padrão X.509v3, a extensão *CRLDistributionPoint* permite indicar onde esta LCR pode ser obtida.

O terceiro passo consiste nas entidades finais acessando este repositório para a validação dos certificados.

A emissão de uma LCR completa não é uma ótima solução [ANK2002] em termos de espaço e tempo. Uma LCR pode se tornar muito grande conforme os certificados vão se tornando inválidos. Este crescimento requer mais espaço de armazenamento, um aumento de banda para o envio da LCR para o local de armazenamento público e mais tempo para as entidades finais processarem os certificados. Para resolver estes problemas, vários mecanismos para particionar LCRs foram definidos como Distribution Points, LCRs Delta, LCRs Indiretas e até um protocolo para verificação on-line de certificados revogados chamado OCSP [ANK2002]. Todos os mecanismos citados, porém, necessitam um suporte à criação de uma LCR completa como uma funcionalidade básica para seu funcionamento, por isso, este método de geração será adotado para a revogação dos certificados no âmbito da ICP-Pro, devendo futuramente ser expandido para algum dos outros métodos citados.

A figura 5.10 apresenta o modelo de utilização da LCR com os seus passos.

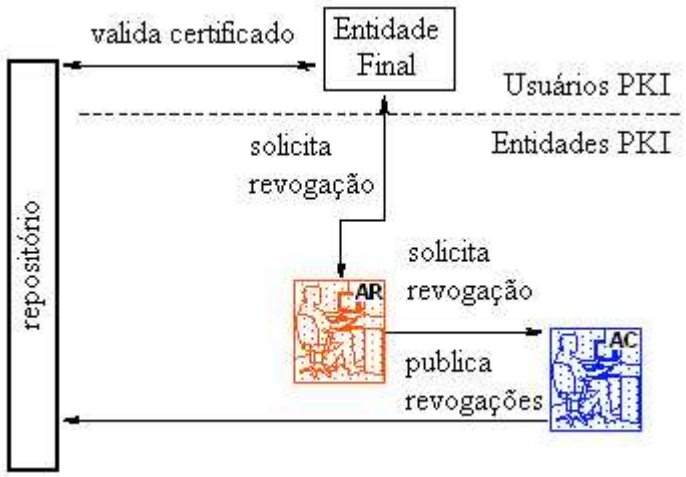

FIGURA 5.10 - modelo de utilização da LCR

5.6.2.8 Armazenamento da LCR

Para o armazenamento da Lista de Certificados Revogados será utilizado o LDAP. A melhor forma encontrada para o armazenamento desta lista foi com a criação de um nova classe, denominada diretoAC, para o armazenamento destes dados junto ao certificado da AC e de sua chave pública.

O esquema utilizado pelo DiretoGNU precisa ser estendido, com a inclusão de novos atributos, para que possa armazenar estas informações.

A extensão do esquema do DiretoGNU prevê a criação de uma nova classe, conforme modelo abaixo:

```
objectclass ( 1.3.6.1.4.1.15509.1.3
      NAME  'diretoAC'
      DESC  'objeto temporário para o direto. FIXME'
      SUP   top
      STRUCTURAL
            MUST (
                  objectClass $ cn $ uid $ empresa
                 )
            MAY  (
                  diretoACCertificado $ diretoACRevocationList $
                  diretoCAPublicKey
                 )
            )
```

Os três atributos que foram estendidos no esquema já utilizado pelo DiretoGNU precisam ser definidos, conforme segue:

```
attributetype ( 1.3.6.1.4.1.15509.0.76
        NAME 'diretoACCertificado'
        DESC 'usado para armazenar o certificado da AC. FIXME'
        EQUALITY caseIgnoreMatch
```

```
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.8
        )

attributetype ( 1.3.6.1.4.1.15509.0.77
        NAME 'diretoACRevocationList'
        DESC 'usado para armazenar a lista de certificados
revogados da AC. FIXME'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.9
        )

attributetype ( 1.3.6.1.4.1.15509.0.77
        NAME 'diretoACPublicKey'
        DESC 'usado para armazenar a chave pública da AC. FIXME'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.6
   )
```

5.6.2.9 Estrutura da LCR

Nesta seção, são estudados todos os campos da LCR, e quais os valores serão gravados nestes campos. Os campos, sua descrição e o valor para o Direto estão apresentados na tabela 5.5.

TABELA 5.5 – campos da LCR do Direto

Campos			Definição	Valor para o Direto
tbsCertList	version		Indica a versão da LCR	2
	signature		contém o identificador de algoritmo utilizado para assinar a lista	gravada quando da configuração da AC
	issuer		identifica a entidade que assinou e emitiu a LCR	DN da AC-Pro
	thisUpdate		data de emissão da LCR	data de início da gravação dos certificados revogados
	nextUpdate		data em que a próxima LCR será emitida. A próxima LCR poderá ser criada antes desta data, mas nunca depois	thisUpdate + prazo de divulgação da LCR, gravada quando da configuração da AC
	revokedCertificates	userCertificate	numero de série do certificado revogado	número serial que identifica o certificado
		revocationDate	data da revogação do certificado	data atual
		CrlEntryExtensions	conjunto de extensões que permitem que as ACs transportem informações adicionais para cada revogação	conforme definido na tabela 5.6
	crlExtensions		extensões adicionais para o tratamento da LCR	
signatureAlgorithm			o mesmo que gravado em signature	o mesmo que gravado em signature
signatureValue			assinatura da lista	assinatura da lista

O campo crlEntryExtension é um conjunto de extensões que permitem que a AC-Pro transporte informações adicionais para cada certificado revogado. As extensões existentes são as contidas na tabela 5.6.

TABELA 5.6 – crlEntryExtension para a LCR do Direto

Extensão	Definição	Valor para o Direto		Crítica
ReasonCode	identifica a razão para a revogação do certificado. Embora seja um campo não crítico, seu uso é recomendável	unspecified	0	não
		keyCompromise	1	
		cACompromise	2	
		affiliationChanged	3	
		suspensed	4	
		cessationOfOperation	5	
		certificateHold	6	
		removeFromCRL	8	
HoldInstructionCode	essa extensão suporta a suspensão temporária de um certificado	none	1	não
		callissuer	2	
		reject	3	
InvalidityDate	contém um valor de data e hora que mostra quando ocorreu um comprometimento suspeito ou conhecido da chave privada	não será utilizado pelo Direto		não
CertificateIssuers	CertificateIssuers: identifica o nome do emissor do certificado associado a uma LCR Indireta	não será utilizado pelo Direto		não

Uma LCR também fornece extensões que se referem a toda a LCR e não a um certificado específico. As extensão estão definidas, juntamente com sua descrição e uso no Direto, na tabela 5.7.

TABELA 5.7 – Extensões da LCR

Extensão	Definição	Valor para o Direto	Crítica
Authority Key Identifier	provê uma maneira de identificar a chave pública correspondente à chave privada utilizada para assinar a CRL		sim
Issuer Alternative Name	permite que identificações adicionais do usuário associadas ao emissor da CRL. Estas identificações podem ser: endereço de e-mail, uma entrada DNS, um endereço IP e uma URI		não
CRL Number	contém um número seqüencial que identifica a LCR de determinada AC. Através deste campo pode-se facilmente saber se existe uma versão mais atual da LCR	número seqüencial que identifica a LCR	sim
Delta CRL Indicator	identifica se a LCR é uma LCR Delta.		sim
Issuing Distribution Pointer	identifica o ponto de distribuição para uma determinada LCR e indica se a lista abrange apenas a revogação de certificados de usuários, ou de certificados de Acs		sim

5.6.2.10 Publicação da LCR

Depois que uma LCR [BUR2002] é criada e assinada digitalmente, ela pode ser distribuída livremente através da rede ou armazenada em um diretório da mesma maneira como os certificados.

As ACs emitem periodicamente LCRs em agendas que variam desde algumas horas até algumas semanas. Uma nova LCR é emitida havendo ou não novas informações, assim, as partes verificadora sempre sabem que a LCR recebida é atual.

Uma organização precisa determinar como publicar uma LCR, informando qual ponto de distribuição irá utilizar. O tipo de LCR a ser usada também necessita ser decidida. Por exemplo, uma organização pode publicar listas diferentes de acordo com a severidade da revogação. O tempo de divulgação entre as várias listas criadas é importante, assim como o método de divulgação. Existem 3 métodos de se publicar uma LCR.

- método Pull: neste método [KUH2001][CON1998][JOH2000] as aplicações são levadas a periodicamente checar repositórios conhecidos para verificar a última alteração da LCR, atualizando seu próprio repositório. Neste método talvez o certificado tenha sido revogado, mas ainda não tenha sido publicado no repositório. O método Pull permite ao repositório AC transferir a reputação de exposição de risco para a parte confiável com respeito a aceitação de um certificado;

- método Push: neste método [KUH2001][CON1998][JOH2000] a LCR é distribuída cada vez que uma nova LCR é criada. Neste método, a própria distribuição torna-se um problema, já que algumas aplicações podem não estar disponíveis no momento da distribuição. Neste método, a responsabilidade de aceitação do certificado é totalmente da AC já que a LCR pode não ter sido distribuída em tempo hábil;

- verificação on-line: neste método [KUH2001][CON1998][JOH2000] a aplicação executa uma pesquisa on-line em uma AC particular antes de validar um certificado. Este método possui uma série de vantagens inclusive a de permitir que a informação da revogação esteja disponível já na hora. A desvantagem é que a Procergs necessitará manter um servidor seguro o tempo todo e terá que assinar digitalmente cada pesquisa, o que cria um alto tráfego de processamento.

O método escolhido para o Direto é o método Pull, já que a LCR será publicada em um diretório público, obedecendo o prazo de divulgação configurado junto a AC, na forma de um arquivo exportado chamado LCR_Pro.DER. Este arquivo estará indicado na extensão LCR Distribution Point, na forma de uma URI.

A LCR será divulgada no formato de codificação DER (*Distinguished Encoding Rules*), que garante uma codificação única para cada valor ASN.1. O DER é geralmente utilizado quando se pretende garantir a compatibilidade da codificação ASN.1 com implementações heterogêneas. É utilizado após a descrição do certificado, no momento do transporte, quando os dados são codificados de acordo com o DER de modo que os

mesmos possam ser armazenados e transferidos independentemente da plataforma de hardware e software.

O prazo de divulgação da lista de certificados se dá conforme a configuração, porém, algumas razões críticas podem alterar esta data fazendo com que a lista tenha sua data de publicação adiantada. A tabela 5.8 contém as razões da revogação e os prazos para a divulgação das LCRs.

TABELA 5.8 – razões de revogação e seus prazos para divulgação

Razão	**prazo para divulgação**
Não especificada	obedece a data da próxima divulgação
Comprometimento de chave	imediata
Comprometimento da chave da AC	imediata
Troca de nome	obedece a data da próxima divulgação
Suspensão	obedece a data da próxima divulgação
Cessação de operação	obedece a data da próxima divulgação
Remover da CRL	obedece a data da próxima divulgação

5.6.2.11 Buscar chaves privadas de usuários

Sempre que um usuário necessitar assinar ou cifrar algum documento através do Direto, ele inicia uma sessão segura com a AR para solicitar a chave, enviando a identificação do usuário. A AR, após validar se o usuário é um usuário válido no banco de dados do Direto, através de uma sessão segura busca a chave privada do usuário no banco de dados PostgreSQL. Através desta mesma sessão segura, envia a chave para o usuário.

5.6.3 Autoridade Registradora da Procergs (AR-Pro)

A AR irá atuar como uma interface entre a AC-Pro e o usuário final recebendo requerimentos dos usuários, autenticando-os e então enviando as requisições para a AC-Pro. Após receber uma resposta da AC-Pro, a AR-Pro notifica a entidade final dos resultados.

Uma vantagem da criação da AR-Pro é o aumento de segurança da AC-Pro. O risco de quebra de segurança de uma AC é fortemente reduzida já que uma pequena quantidade de pessoas possui acesso a AC-Pro, além da AR-Pro.

Como um exemplo, pode-se imaginar a Secretaria da Saúde, que possui postos de trabalho em várias cidades do Estado do Rio Grande do Sul. Cada posto de trabalho terá uma AR instalada, que fará a autenticação dos usuários do seu posto de trabalho. As tarefas pertinentes a AR-Pro são detalhadas a seguir.

5.6.3.1 Verificação do par de chaves do usuário

Quando o usuário requisita a geração de um certificado, o sistema verifica se o usuário já possui um par de chaves geradas para ele. Se não possuir, o sistema deve gerar o par de chaves. Após a geração do par de chaves, a AR-Pro pode então anexar a chave

pública do usuário na requisição do certificado e enviá-la para que a AC-Pro conclua o processo de geração do certificado.

5.6.3.2 Autenticação do Usuário

Quando o usuário requisita um certificado, a AR-Pro deve verificar se todos os dados necessários para a criação estão disponíveis em seu registro no diretório. A falta de algum das informações fará com que o certificado fique à espera de uma confirmação. O administrador do sistema deverá avisar o usuário, enviando um e-mail, sobre quais as informações estão ausentes ou incorretas. Um usuário só é autenticado quando todas as informações para a criação do certificado estiverem presentes e válidas.

As informações de usuário que devem estar presentes no diretório são: *matricula*, *dataadmissao*, *setor*, *cargo*, *datademissao* (esta data deve estar nula), *sexo*, *cpf* e *identidade* que correspondem as informações de matrícula, data de admissão, setor, cargo, data de demissão, sexo, número do CPF e número da carteira de identidade do usuário.

5.6.3.3 Edição de Requisição

Esta opção será utilizada sempre que o administrador do sistema precise alterar alguma informação necessária em uma requisição de certificado, ou quando for necessário, por falta dos requisitos necessários ao solicitante, alterar o tipo de certificado a ser gerado.

5.6.3.4 Aprovação de Requisição

Esta opção será utilizada quando o administrados do sistema tiver validado todos os dados necessários para a geração de um certificado para um usuário, atestando que as informações são válidas e que o tipo de certificado solicitado pelo usuário pode ser gerado para ele.

Nesta opção, a requisição, na forma de *template* é encaminhada à AC-Pro, para a geração do certificado.

5.6.3.5 Rejeição da requisição

Esta tarefa é realizada quando o administrador do sistema não conseguir validar todas as informações necessárias para garantir a identidade do usuário solicitante do certificado, ou quando o certificado solicitado por ele não for do tipo que ele pode receber, ou mesmo quando o tempo para o preenchimento das informações tiver espirado.

Nesta opção, a requisição é armazenada no Registro de Requisições e o usuário deverá ser avisado sobre a rejeição.

5.6.3.6 Envio de requisição para a AC-Pro

O envio de um certificado para a AC-Pro pode se dar de duas formas:

- usuário solicita um certificado a AR-Pro, porém seus dados não estão completos. A AR-Pro de posse de todas as informações necessárias aprova a requisição e então ela é enviada à AC-Pro;

- um usuário solicita um certificado. Se todos os dados estiverem completos a requisição é enviada automaticamente para a AC-Pro.

5.6.3.7 Envio de resposta ao usuário

Após a AC-Pro criar e assinar o certificado do usuário, a AR-Pro é avisada do término do processo. Neste momento, a AR-Pro deve gerar uma informação para o usuário, avisando da criação do certificado, além de atualizar a sua lista de Registro de Requisições.

5.6.3.8 Manutenção dos registros de requisições

Todas as requisições de Certificados deverão ficar arquivadas, tendo sido aprovadas ou anuladas. Para um correto histórico destas requisições, será mantido, além dos dados da requisição, o *status* da requisição, a data de aprovação ou anulação, o motivo da rejeição (quando for o caso), e o nome do administrador do sistema que rejeitou ou aprovou a requisição. Quando a requisição for aprovada sem a intervenção de um usuário, o campo de nome do administrador irá conter a informação AC-Pro. Para manter estes dados, será criada uma nova tabela, no banco de dados, chamada AR_solicitacoes. As colunas necessárias para a manutenção desta tabela encontram-se na figura 5.11.

coluna	**descrição**	**valores válidos**	
usuario	usuário que fez a solicitação		
data	data da alteração da solicitação		
tipo	tipo de solicitação	criação	C
		revogação	R
estado	estado atual da solicitação	processo não iniciado	P
		incompleta	I
		enviado à AC-Pro	E
		certificado gerado	G
		certificado Ok	K
		DPC não aceita	D
aceita_dpc	informa se o usuário aceitou a DPC da ICP-Pro	não aceita	N
		aceita	S
motivo_revogacao	motivo do pedido de revogação do certificado		
cargo_solicitante	cargo do usuário que solicitou a revogação		
nro_certificado	número serial do certificado a ser revogado		
mot_rejeicao	motivo da rejeição da requisição		
nm_adm	nome do administrador que aprovou ou rejeitou a requisição		

FIGURA 5.11 – definição da tabela AR_solicitacoes

5.6.3.9 Controle do tempo para usuário completar requisição

O usuário terá um tempo hábil para a entrega das informações que faltam para a aprovação da requisição. A AR-Pro fica responsável por controlar este prazo, de acordo com o que for configurado na tela de configurações da AC-Pro, no campo Prazo para Completar Requisição. Se neste prazo o usuário não tiver entregue todas as informações, a requisição será anulada, o registro de requisição da AR-Pro será atualizado com o *status* de "Tempo Expirado", e o usuário receberá um aviso da AR-Pro sobre esta anulação.

5.6.3.10 Avisar o usuário sobre Política e solicitar confirmação

Sempre que um usuário fizer a solicitação de um certificado, a AR-Pro deverá apresentar a Política de Certificação da ICP-Pro para a sua aprovação. Se o usuário não aceitar esta Política, o usuário será mantido como Pendente, até que o usuário a aprove.

Após passado o prazo para completar requisição, esta será anulada.

5.6.3.11 Lista de certificados revogados e expirados

Quando solicitado por um usuário, a AR-Pro deverá mostrar uma lista dos certificados que foram revogados ou expirados, com seus motivos, obedecendo a filtros específicos, como data e motivo.

5.7 Definindo políticas de segurança

Um dos primeiros passos que uma organização tem que considerar quando soluções de segurança vão ser introduzidas é definir um possível conjunto de políticas de segurança.

5.7.1 Declaração de Práticas de Certificação (DPC)

Uma DPC é definida pela American Bar Association [AME1995] como "uma declaração das práticas que a autoridade certificadora emprega na emissão de certificados". Isto é, um documento que explica qual é exatamente o escopo de um certificado e qual o degrau de confiança que pode-se ter em relação aquela AC. Ela permite ao usuário do certificado tomar a decisão sobre quando confiar ou não nos certificados criados por aquela AC. A RFC2527 contém uma lista de tópicos que necessitam ser cobertos em uma política de certificados.

A montagem da DPC para a ICP-Pro ajudou a validar o modelo, já que apresenta todo o processo, desde as obrigações da AC-Pro até as responsabilidades do usuário dos certificados. O texto completo da DPC da ICP-Pro está no Anexo A deste trabalho.

A DPC da ICP-Pro será apresentada ao usuário, para leitura e aceitação, sempre que ele solicitar a criação de um certificado. O certificado só é gerado se o usuário concordar com todas as cláusulas contidas nesta DPC.

5.8 Interfaces

Uma fase importante do modelo aqui apresentado trata da questão da interface da ICP-Pro com os usuários, sejam os administradores da AC-Pro, da AR-Pro ou mesmo dos usuários finais, solicitantes dos certificados.

A tela principal do usuário terá incluída a opção de Segurança. Esta opção terá como itens adicionais: Solicitar Certificado, Revogar Certificado e Renovar Certificado (disponível apenas se configurado junto à AC que a renovação terá a intervenção do usuário).

O Direto terá sua tela de envio de mensagens alterada, para que possa conter as opções de cifrar e/ou assinar mensagens. Por *default*, esta opção virá desmarcada, sendo utilizada apenas se o usuário solicitar, escolhendo uma das opções ou ambas.

A tela de recepção de mensagens também será alterada. Na lista de mensagens recebidas serão incluídos dois novos ícones, para marcação das mensagens. Um ícone define uma mensagem recebida cifrada. Outro ícone define uma mensagem recebida assinada.

Quando o usuário abrir uma mensagem marcada como cifrada, o sistema se encarrega de buscar a chave privada do usuário (armazenada cifrada no banco de dados), decifrá-la e utilizá-la para decifrar a mensagem.

Quando o usuário abrir uma mensagem marcada como assinada, o sistema se encarregará de verificar se o certificado é válido. Se o certificado não for válido, a mensagem será sinalizada, indicando que a assinatura não pode ser validada e o usuário será avisado sobre o erro. A mensagem poderá, mesmo assim, ser visualizada, porém, acreditar na autenticidade da mensagem ficará a cargo do usuário.

O modelo proposto terá como válido todo o certificado que tiver como emissor a própria Procergs. A aceitação de uma assinatura válida, quando o emissor do certificado for uma outra AC, que não a AC-Pro, necessita da inclusão de outras funcionalidades ao modelo. O armazenamento das autoridades certificadoras confiáveis deverá ser criado para a validação das assinaturas.

5.9 Considerações Finais

O objetivo principal deste capítulo foi apresentar um modelo de infra-estrutura de chaves públicas para a Procergs, para uso pelos usuários do Direto, de modo a garantir a privacidade, a integridade, o não-repúdio e a autenticidade das mensagens enviadas e recebidas por seus usuários.

O modelo proposto é apenas uma base para uma infra-estrutura maior, já que trata apenas dos e-mails dos usuários cadastrados no Direto. Por ser um modelo aberto e flexível,

que será totalmente desenvolvido dentro da própria estrutura da Procergs, este modelo permite que, em um futuro próximo, a Procergs possa vir a se tornar uma Autoridade Certificadora, podendo então garantir a segurança de outros usuários, diferentes documentos ou aplicativos.

A partir da especificação deste modelo, foi implementado um protótipo para assistir o usuário no ambiente Direto, capaz de automatizar as etapas previstas e que pudesse ser utilizado para a validação do mesmo. O mesmo será descrito no próximo capítulo.

6 Protótipo da Infra-estrutura de Chaves Públicas

Após a especificação do modelo de uma infra-estrutura de chaves públicas para utilização na segurança das mensagens trafegadas pelo software Direto da Procergs, este capítulo vai apresentar e detalhar o protótipo implementado para automação de cada uma das entidades e etapas descritas no modelo.

Inicialmente serão relacionadas as principais características pretendidas para o protótipo. Em seguida, serão apresentados o ambiente e as ferramentas utilizadas para o seu desenvolvimento. Finalmente, serão apresentadas a estrutura de implementação e a interface disponibilizada junto ao Direto para a criptografia, assinatura e validação dos certificados.

6.1 Considerações Iniciais

Após o estudo de algumas ferramentas já desenvolvidas e de domínio público, e a opção por não utilizá-las, tomou-se o caminho de fazer-se a implementação utilizando-se a própria linguagem Java. Para isso, fez-se uma procura na literatura sobre ferramentas ou classes Java que atendessem a esse propósito. Nessa pesquisa, foi encontrado, primeiramente a ferramenta KeyTool. Esta ferramenta está descrita no capítulo 3, assim como está lá detalhado o porque de sua utilização apenas para a criação de certificados para testes.

Logo após a determinação de que o keytool seria utilizado apenas para testes, encontrou-se o pacote Bouncy Castle, que é totalmente desenvolvido em Java e possui todos os métodos necessários para o controle do ciclo de vida dos certificados, assim como os métodos para envio e recebimento de e-mails criptografados e/ou assinados. O pacote Bouncy Castle está descrito no capítulo 3.

6.2 Armazenamento dos Dados

O local e a forma como são armazenados os dados é sempre um ponto importante na implementação de um protótipo. A seguir, a definição do armazenamento dos dados utilizados pela ICP_Pro.

6.2.1 Arquivo de Configurações

Dois arquivos de configuração serão utilizados. O arquivo AC.CONFIG e o arquivo AR.CONFIG.

O arquivo AC.CONFIG, mostrado na figura 6.1 tem as configurações que o usuário informa quando da configuração da AC. Estas configurações são referentes aos métodos de criptografia que serão utilizados para a criação das chaves e dos certificados.

```
#configuracoes referentes
# a criacao das chaves dos usuarios
algoritmo_chave=RSA
tamanho_chave=512

#configuracoes referentes
#a criacao dos certificados dos usuarios
algoritmo_assinatura=SHA1withDSA
prazo_validade=1
renovacao=AVISAR
prazo=1
```

FIGURA 6.1 - exemplo de arquivo configurado pelo usuário

6.2.2 LDAP

Conforme definido no capítulo 5, o esquema de LDAP existente hoje no Direto precisa ser alterado para que possa armazenar os dados da AC e os dados de certificados e chaves públicas dos usuários. A Lista de Certificados Revogados também será armazenada no LDAP.

Para a alteração do esquema, o arquivo Direto.schema foi alterado manualmente, incluindo os novos atributos e objetos. Após a alteração, o arquivo em uso pelo Direto foi substituído por este novo arquivo.

6.2.3 PostgreSQL

O banco de dados PostgreSQL vai armazenar a chave privada dos usuários, na tabela USUARIO, de forma cifrada. Esta forma de armazenamento foi escolhida porque o acesso ao banco de dados tem a capacidade de ser mais segura. A implementação de segurança do banco de dados não faz parte do escopo deste projeto.

A tabela AR_SOLICITACOES, mostrada na figura 6.2, será a tabela responsável por manter os dados históricos das solicitações dos usuários. Nesta table é mantido o estado atual de cada requisição e a data em que ele foi alterado.

FIGURA 6.2 – Modelagem das tabelas do PostgreSQL

Para a criação das tabelas no PostgreSQL foi necessário converter o modelo para uma sintaxe SQL, conforme mostrado aqui:

```
create table AR_SOLICITACOES (
    empresa varchar(60) not null,
    usuario varchar(60) not null,
    data date not null,
    tipo char(1) not null,
    estado char(1) not null,
    motivo_revogacao varchar(500) null,
    cargo_solicitante varchar(60) null,
    nro_certificado numeric(20) not null,
    motivo_rejeicao varchar(200) null,
    nome_adm varchar(60) null,
    primary key (empresa, usuario));

create table USUARIO (
    empresa varchar(60) not null,
    usuario varchar(1) not null,
    chave_pivada varchar(2400) not null,
    primary key (empresa, usuario));

create unique index IDAR_SOLICITACOES
    on AR_SOLICITACOES (empresa, usuario);

create unique index IDUSUARIO
    on USUARIO (empresa, usuario);
```

Estas alterações foram submetidas ao banco de dados *diretodb*, que é o banco utilizado para armazenar as informações do módulo de agenda.

6.3 Criação da AC-Pro

A AC-Pro é um módulo independente, criado para ser instalado em uma máquina isolada, abaixo de um firewall interno. O único componente da ICP que tem acesso a AC-Pro é a AR-Pro, através de um túnel SSL seguro.

Ao entrar no módulo AC o sistema tem acesso a dois menus, o menu com as opções de configuração da própria AC e o menu com as opções disponíveis para o gerenciamento dos certificados dos usuários, como mostra a figura 6.3.

FIGURA 6.3 – Interface do módulo AC

Cada item do menu AC-Pro tem uma função específica, com a finalidade de tornar a certificadora AC da Procergs ativa e em funcionamento. Estes itens serão descritos a seguir.

6.3.1 Submenu AC

Os passos a serem seguidos para a configuração da AC Raiz da Procergs são apresentados no submenu AC, na ordem em que devem ser executados. O submenu AC pode ser visto na figura 6.4 e seus itens são detalhados a seguir.

FIGURA 6.4 – Submenu AC

6.3.1.1 Inicializa AC

O item Inicializa AC tem a finalidade de gerar o par de chaves (chave pública e chave privada) do componente AC. Como o componente AC e suas chaves é o ponto básico do sistema de segurança da ICP-Pro, o algoritmo de criptografia e o tamanho destas chave criadas por ela foram definidos previamente, não sendo disponível ao Administrador do Sistema a escolha destes parâmetros. Se, com o passar do tempo, surgir um tamanho de chave maior, ou algoritmo mais seguro, todo um processo de recriação de chaves e certificados deverá ser iniciado, onde uma das etapas será alterar o código fonte da AC para que possam ser definidos os novos parâmetros.

O processo de geração do par de chaves irá resultar na criação da chave privada, que é armazenada no arquivo ca_priv.key, e da chave pública, que é armazenada no arquivo ca_pub.key .

Para a criação das chaves, foi utilizado uma classe Java conhecida chamado de KeyPairGenerator. Foram criados dois métodos para a obtenção do algoritmo de criptografia e para obtenção do tamanho de chave, apenas como uma forma de manter o código mais claro. O algoritmo escolhido para a AC foi o algoritmo RSA e o tamanho de chave é de 2048 bits. A codificação utilizada para a criação do par de chaves é mostrada na figura 6.5.

```
private void criaParChaves()
{
        KeyPairGenerator kpg;
        try {
                kpg = KeyPairGenerator.getInstance(config.getAlgCript());
        }
        catch (NoSuchAlgorithmException e)
        {
                JOptionPane.showMessageDialog(
                                null,
                                "Algoritmo invalido!\n"+e.toString(),
                                "Erro",
                                JOptionPane.ERROR_MESSAGE);
                        return;
        }
        kpg.initialize(config.getTamChave());
        KeyPair par = kpg.genKeyPair();
        pub_key = par.getPublic();
        priv_key = par.getPrivate();

        saveKey(priv_key, PRIV_KEY_FILE);
        saveKey(pub_key, PUB_KEY_FILE);
}
```

FIGURA 6.5 – Código utilizado para a criação do par de chaves da AC-Pro

Após a geração do par de chaves o sistema se encarrega de armazená-las. A chave privada é armazenada no mesmo diretório onde estão as configurações do Direto, e a chave pública é colocada no LDAP, para estar disponível quando os usuários necessitarem utilizá-la para verificação da assinatura.

6.3.1.2 Gera Certificado

Neste item é realizada a criação do certificado da Procergs, que é também chamada de AC-Raiz da ICP-Pro. Uma AC-Raiz é a entidade que se encontra no topo de uma hierarquia, provendo confiança para todos elementos que se encontram abaixo dela . Como não existe nenhuma AC em um nível acima, que assine seu certificado, o seu certificado digital é auto-assinado.

Para a criação do certificado é utilizado o método x509v3CertificateGenerator, que pertencente ao pacote Bouncy Castle. Para a criação do certificado, alguns campos devem ser preenchidos, conforme segue:

- SerialNumber: foi criado um método para a obtenção de um número seqüencial dos certificados. A numeração do certificado é armazenada no LDAP, junto à classe diretoAC, atributo diretoACUltimoCertificado Como o certificado da a AC Raiz é sempre o primeiro certificado a ser gerado, seu SerialNumber será 1 (um).

- NotBefore: é a data da criação do certificado;

- NotAfter: é a data da criação do certificado, mais o prazo default determinado para a sua validade. No caso da AC da Procergs, foi determinado um prazo de 10 anos;

- SubjectDN: neste campo é utilizado o DN da Procergs, no formato C=BR, O=Procergs;

- SignatureAlgorithm: o algoritmo de assinatura pré estabelecido para o certificado da AC Raiz foi o SHA1withRSA.

O método desenvolvido para a criação do certificado é demonstrado na figura 6.6.

```
private X509Certificate criaCertAC()
{
        String issuer = CADistName;
        String subject = issuer;

        X509V3CertificateGenerator v3CertGen = new X509V3CertificateGenerator();

    v3CertGen.setSerialNumber(next_cert);
        next_cert = next_cert.add(BigInteger.ONE);
        v3CertGen.setIssuerDN(new X509Principal(issuer));
        v3CertGen.setNotBefore(new Date(System.currentTimeMillis()));
        v3CertGen.setNotAfter(new Date(System.currentTimeMillis()
                        + config.getPrazoVal() * millisPerYear));
        v3CertGen.setSubjectDN(new X509Principal(subject));
        v3CertGen.setPublicKey(pub_key);

        try {
                String alg = config.getAlgAss()+"Encryption";
                v3CertGen.setSignatureAlgorithm(alg);
                X509Certificate cert = v3CertGen.generateX509Certificate(priv_key);
                return cert;
        }
        catch (Exception e)
        {
                JOptionPane.showMessageDialog(
                                null,
                                "Erro ao assinar o certificado da AC!\n"+e.toString(),
                                "Erro!",
                                JOptionPane.ERROR_MESSAGE);
                return null;
        }
}
```

FIGURA 6.6 - Método para Criação do Certificado da AC Raiz

A AC Raiz é a raiz de uma hierarquia de certificados e indica o ponto inicial de uma cadeia de certificados. Cada uma das ACs abaixo desta e as entidades finais deverão possuir acesso à chave pública da AC Raiz. O certificado gerado para a AC-Pro é mostrado na figura 6.7.

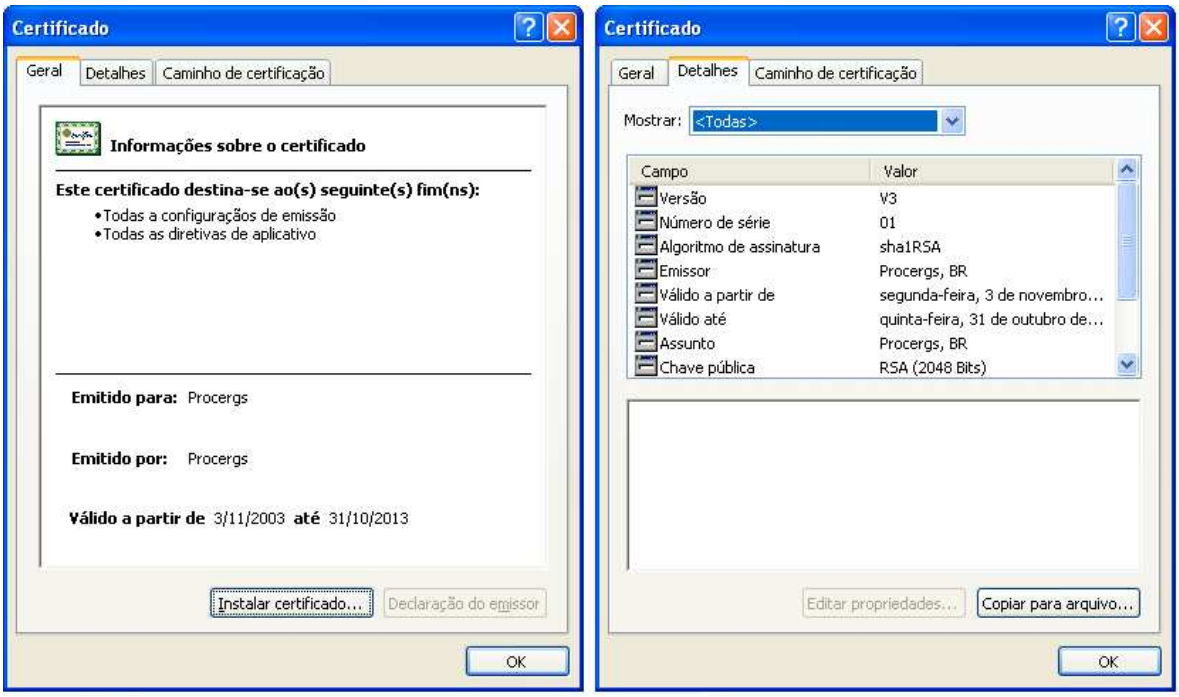

FIGURA 6.7 – O Certificado da AC-Pro

Uma cadeia de certificados [BUR2002] é o método mais comum utilizado para verificar a associação entre a entidade e sua chave pública. Para ganhar a confiança de um certificado, a parte verificadora deve verificar se cada certificado na cadeia está assinado com a chave pública do certificado acima na cadeia. Este encadeamento conta com o fato de as entidades terem acesso a todos os certificado na cadeia. Existem duas formas conhecidas das partes verificadoras obterem estes certificados. Uma maneira é o emissor enviar uma cadeia inteira de certificados quando enviar um certificado, conhecido como modelo push, no qual o remetente empurra a cadeia inteira de certificados ao destinatário e o destinatário pode verificar imediatamente todos os certificados. O modelo pull apenas envia o certificado do remetente e deixa para o destinatário decidir extrair o certificado da AC. Pelo fato de cada certificado conter o nome do emissor, o destinatário sabe onde verificar o certificado. A RFC2459 [HOU1999] tem publicado um algoritmo para a validação de todos os certificados de uma cadeia.

Para o Direto, escolheu-se o método push de obtenção dos certificados. Quando um usuário receber um e-mail que tenha sido assinado por outro usuário do Direto, receberá junto o certificado do remetente e o certificado da AC-Pro. Para este protótipo estamos levando em conta que só existe um nível de confiança para um certificado, que é a própria AC-Pro. O certificado da AC-Pro deverá ser disponibilizado para ser acrescentado ao navegador como uma certificadora confiável para que outros sistemas de e-mail possam fazer a autenticação do remetente de uma mensagem oriunda do Direto. A figura 6.8 mostra o certificado da AC-Pro junto a outros certificados confiáveis já importados.

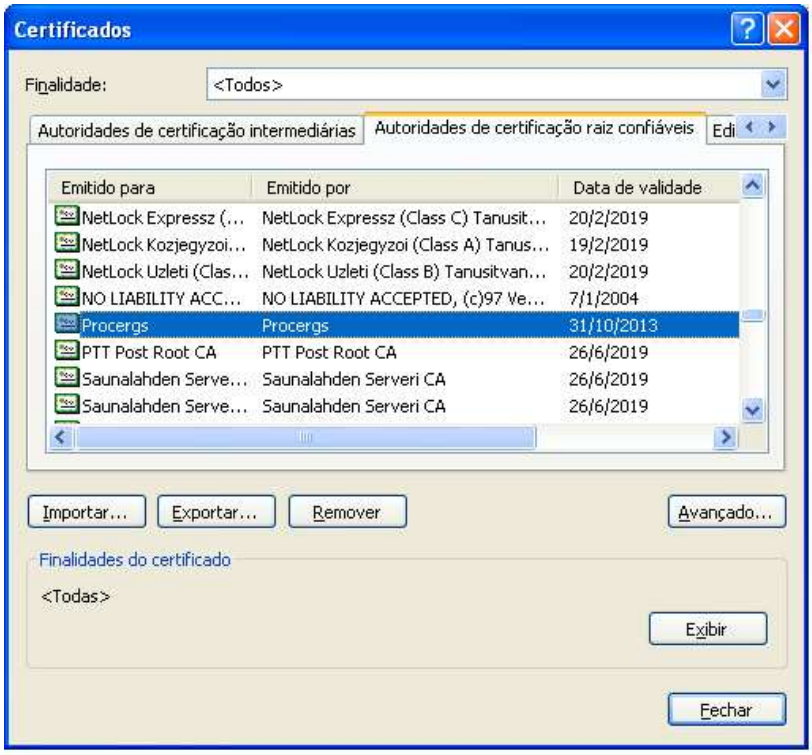

FIGURA 6.8 – Certificado da AC-Pro instalado no navegador.

6.3.1.3 Configuração da AC

No item Configuração da AC, o Administrador da AC fará as configurações necessárias a AC para a geração dos pares de chaves e certificados dos usuários. Estas informações são armazenadas no arquivo AC.CONFIG conforme apresentado no capítulo 5.3.1.

A tela de Configuração pode ser dividida em duas partes distintas, onde em uma parte se configuram os atributos das chaves e na outra se configuram os atributos dos certificados, descritos a seguir.

6.3.1.3.1 Atributos de Chaves

Os atributos necessários à criação das chaves e que podem ser configurados são descritos a seguir e podem ser visualizados na figura 6.9:

- método de criptografia: traz disponível alguns métodos de criptografia para que o administrador possa escolher qual método irá utilizar. Os algoritmos disponíveis são: RSA e DSA;

- tamanho da chave: aqui, o usuário seleciona o tamanho de chave que será utilizado para a criptografia. Os valores disponíveis são: 512, 1024 e 2048.

6.3.1.3.2 Atributos de Certificados

Os atributos necessários à criação dos certificados e que podem ser configurados são descritos a seguir e podem ser visualizados na figura 6.9:

- algoritmo de assinatura: traz disponível alguns métodos de assinatura para que o administrador possa escolher qual deles irá utilizar. Os algoritmos disponíveis são: SHA1WithDSA, MD2WithRSA, MD5WithRSA, SHA1WithRSA;

- prazo de validade: traz disponível alguns tempos de validade do certificado para que o administrador selecione o que mais se adapta ao seu uso de certificados. Este prazo de validade é utilizado para calcular a data de expiração do certificado. Os valores disponíveis são: 1 ano, 2 anos, 5 anos e 10 anos;

- renovação: Esta configuração refere-se a expiração do certificado. Foi definido que o administrador poderá escolher entre renovar automaticamente um certificado ou avisar ao usuário que seu certificado está prestes a expirar. As opções disponíveis são Avisar e Automática. Se o administrador selecionar a opção de Automática, a AR configurada será responsável por proceder ao início do processo de renovação do certificado automaticamente, quando ele estiver prestes a expirar. Se o administrador selecionar a opção de Avisar ao usuário (que será feito através de e-mail), o sistema solicita também o prazo de antecedência com a qual a AR avisará o usuário que seu certificado está prestes a vencer e quais as providências devem ser tomadas para a renovação.

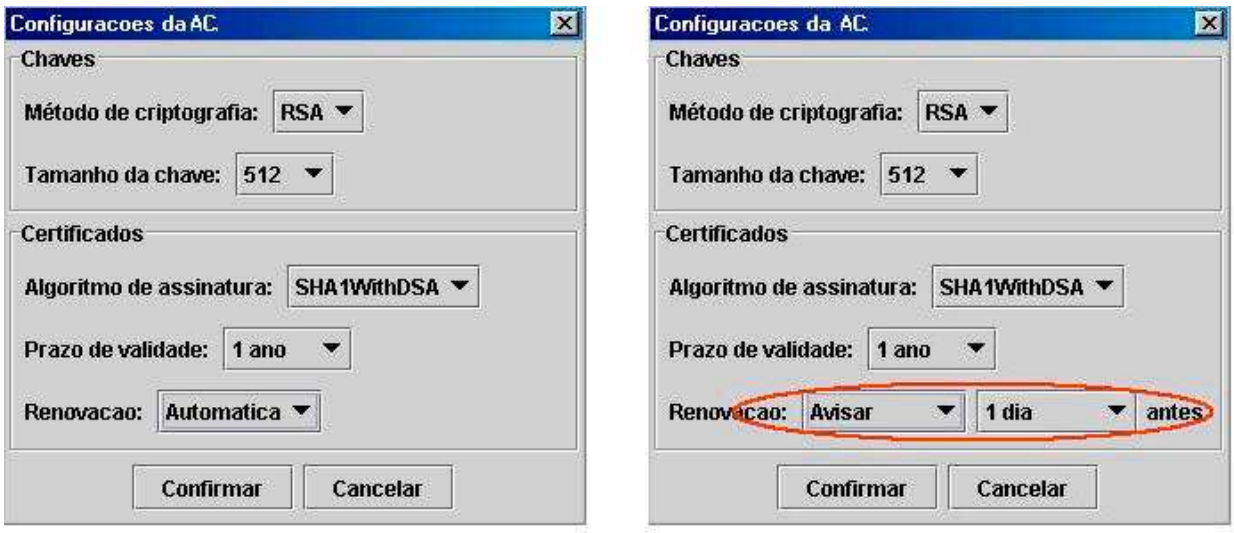

FIGURA 6.9 - Configuração da AC-Pro

6.3.2 Submenu Usuário

Neste submenu, apresentado na figura 6.10, são apresentadas algumas opções que foram definidos para serem utilizados para uso com a finalidade de testar a AC-Pro. Com o desenvolvimento do protótipo optou-se em não desenvolver estas funcionalidades, já que

elas teriam apenas um papel secundário. A inclusão destas opções na AC-Pro teriam duas finalidades, conforme descritas a seguir:

FIGURA 6.10 – Submenu Usuário

6.3.2.1 Solicita Certificado

A opção de solicitação de um certificado foi colocado junto ao módulo AC-Pro com a finalidade de facilitar os testes. Com esta opção, a solicitação e criação de certificados para usuários diferentes, não cadastrados no Direto seria disponibilizada, com a intenção de testar a segurança do sistema. Os testes com os certificados serão descritos no capítulo 7, porém, serão utilizados apenas certificados de usuários que já utilizam o Direto.

6.3.2.2 Lista certificados requisitados

A opção de listar os certificados requisitados seria disponibilizado neste módulo já que o módulo AR, que não é uma entidade obrigatória em uma ICP, sendo uma intermediária entre a AC e o usuário, não terá seu funcionamento desenvolvido, para a finalidade da prototipação. Porém, como foi visto na definição do Modelo, no capítulo 5, o seu uso é necessário, devendo ser implementado. Assim, quando da implementação do projeto como um todo, a opção fará parte do módulo AR-Pro.

6.4 Alteração do Direto

O módulo de correio do Direto funciona como um serviço de correio eletrônico normal, com as facilidades da Internet. Neste módulo o usuário pode ler, enviar, receber e encaminhar mensagens para qualquer contato, dentro ou fora da empresa. Especificamente neste módulo o protótipo desenvolvido neste trabalho será acoplado para garantir a segurança das mensagens trafegadas pelo Direto. A interface do correio está ilustrada na figura 6.11

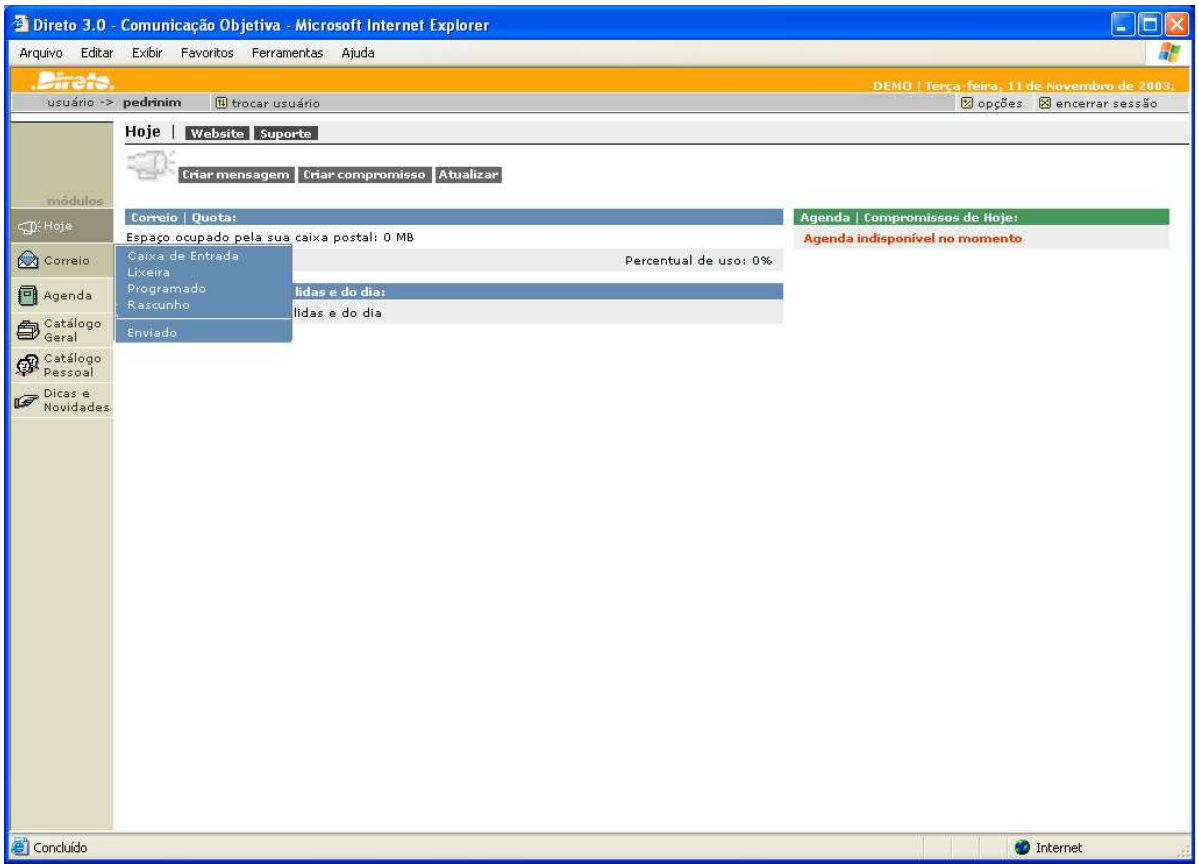

FIGURA 6.11 – Interface do correio do Direto

Um novo item foi adicionado aos itens já existentes no módulo de correio, chamado de Segurança. Neste item, foram adicionadas todas as opções necessários para que o usuário possa manter a segurança das suas mensagens no ambiente do Direto. Estas opções, conforme mostra a figura 6.12, serão descritas a seguir.

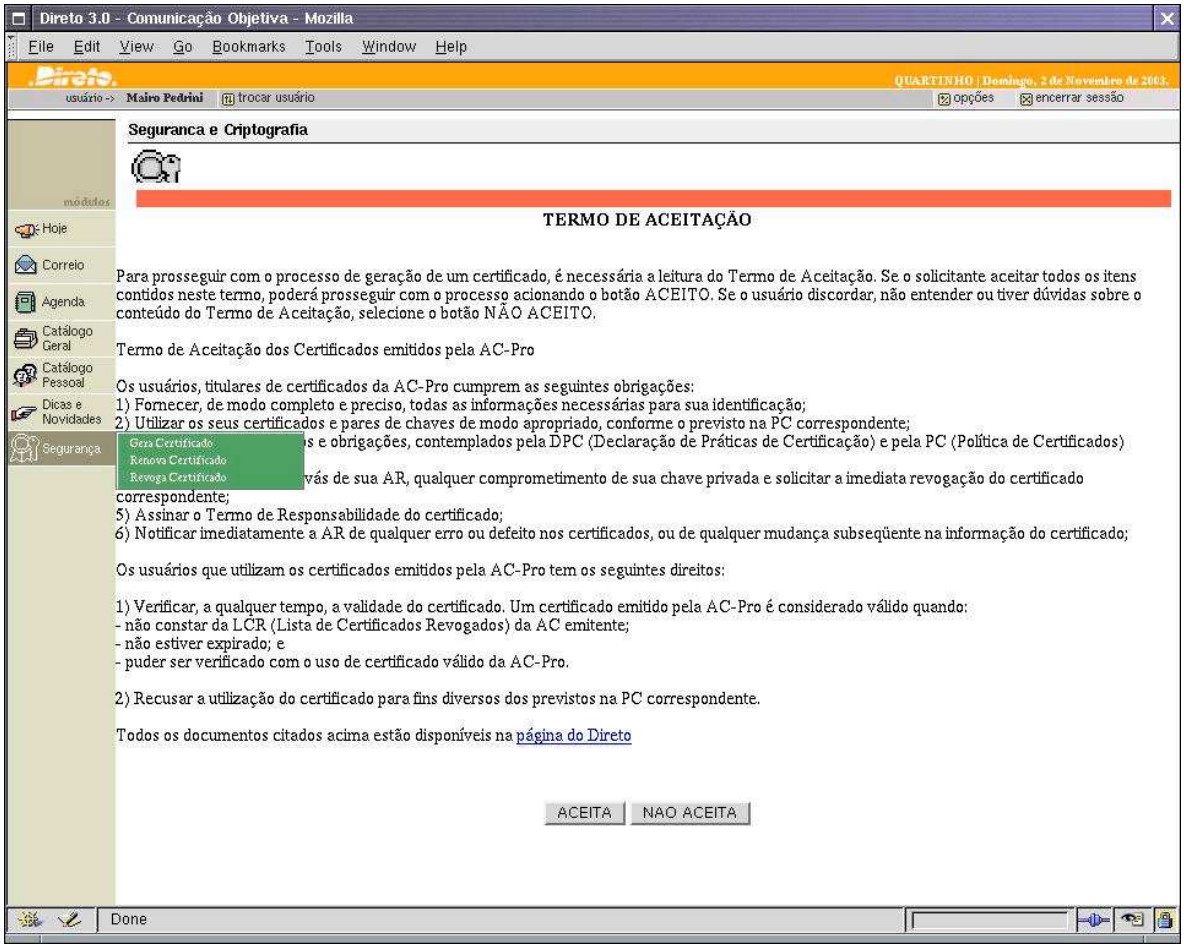

FIGURA 6.12 – Novo item adicionado ao Direto

6.4.1 Gera Certificado

Nesta opção estava previsto o desenvolvimento do item 5.5.1.7 – Criação dos Certificados, exatamente como foi definido em seu Estudo de Caso. Porém, algumas adaptações foram realizadas para manter o produto com uma interface mais amigável e com mais facilidade de uso.

Quando o usuário solicita um certificado, a primeira ação tomada pelo sistema será a de apresentar um Termo de Aceitação, que mostra os direitos e obrigações do usuário em relação a emissão dos certificados da Procergs. Optou-se em não mostrar toda a Declaração de Práticas de Certificação da AC-Pro (conforme Anexo A, neste trabalho) por ser um texto extenso e de leitura cansativa. Extraiu-se, então, as partes fundamentais, que são exatamente as partes que falam sobre o relacionamento do usuário com seu certificado. No termo apresentado existe uma referência a esta DPC e um *link*, para que o usuário possa acessá-la, se assim o desejar. A apresentação do termo ao usuário como primeiro passo foi implementado por julgar-se que, se o usuário não aceita este termo, não existe motivo para que seja gravado um registro na tabela AR_SOLICITACOES. O usuário tem duas opções, conforme mostra a figura 6.13, que é aceitar ou não aceitar o Termo de Aceitação. Se o

usuário optar por não aceitar o termo, o processo é encerrado e uma mensagem de Cancelamento de Operação é apresentada.

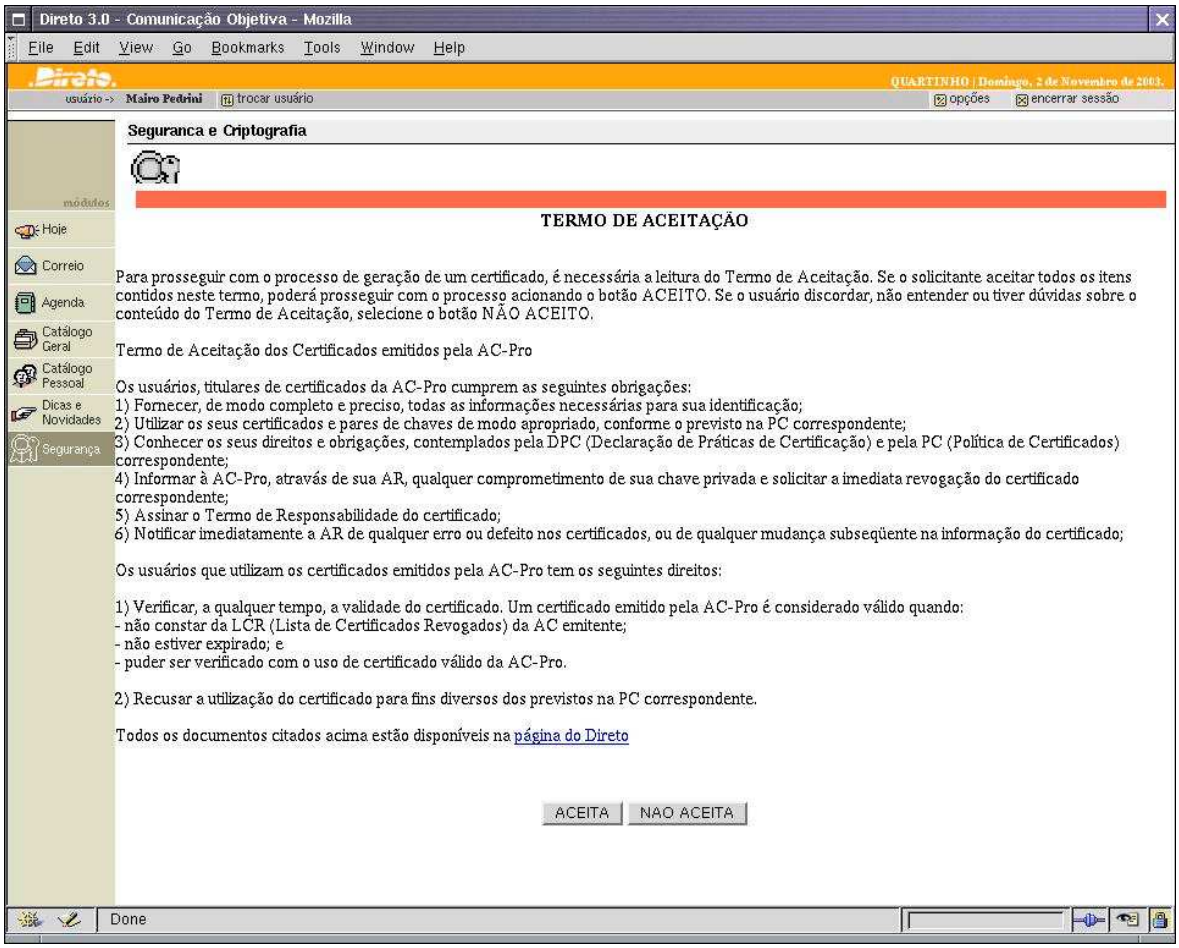

FIGURA 6.13 – Termo de Aceitação do Certificado

Se o usuário optar em aceitar o Termo de Aceitação, o processo de geração do certificado continua, conforme os passos que são definidos a seguir.

O início do processo de geração do certificado de um usuário se dá com a gravação da solicitação na tabela AR_SOLICITACOES, que reside junto a AR-Pro. A definição dos valores para cada campo da tabela se dá de acordo com o que segue:

- empresa: neste campo é gravado o valor digitado pelo usuário no campo Organização, quando ele realizou o login no sistema Direto, conforme mostrado na figura 6.14;

- usuário: aqui é gravada a identificação do usuário, que foi digitada no campo Identificação quando da entrada no sistema Direto, conforme mostrado na figura 6.14;

- data: data atual do sistema

- tipo: para a solicitação de criação de um certificado o valor gravado nesta coluna é o <C>;

- estado: neste momento o estado da solicitação é <P>, que indica um processo ainda não iniciado;

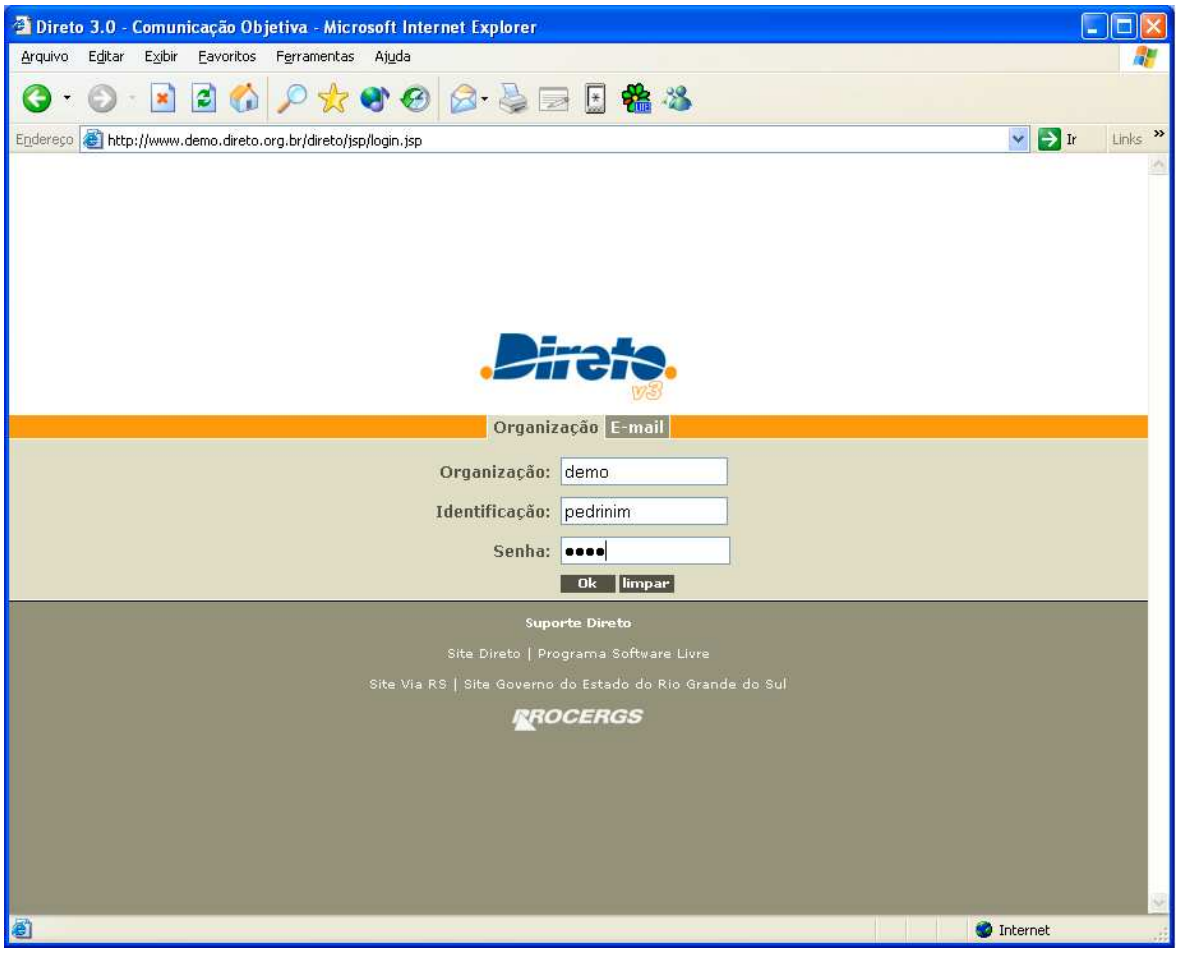

FIGURA 6.14 – Tela de login do sistema Direto

Segundo definido no modelo (capítulo 5), após o registro ser gravado na tabela de solicitações, a AR-Pro estaria apta a prosseguir com o processo, validando os dados do usuário através das informações gravadas no LDAP. Porém, como a AR-Pro não será implementada, a validação dos dados é feita pela AC-Pro, na seqüência. Os dados obrigatórios para a geração de um certificado e que devem estar preenchidos no LDAP foram definidos como: matricula, dataadmissao, setor, cargo, datademissao (esta data deve estar nula), sexo, cpf e identidade. Embora o LDAP tenha estes atributos definidos, o Direto não possui uma interface para o cadastro destas informações. A alteração necessária ao Direto para o cadastro destas informações não faz parte do escopo deste projeto.

A validação dos dados do usuário pode retornar uma resposta válida ou não válida. Se as informações estiverem inconsistentes retornando uma resposta não válida, a tabela AR_SOLICITACOES é alterada, o campo estado passa a ter o valor <D> de dados incompletos. O sistema então envia um e-mail para o usuário, conforme figura 6.15, informando os dados do usuário e qual(ais) do(s) dado(s) estão incompletos. O processo de geração é encerrado aqui.

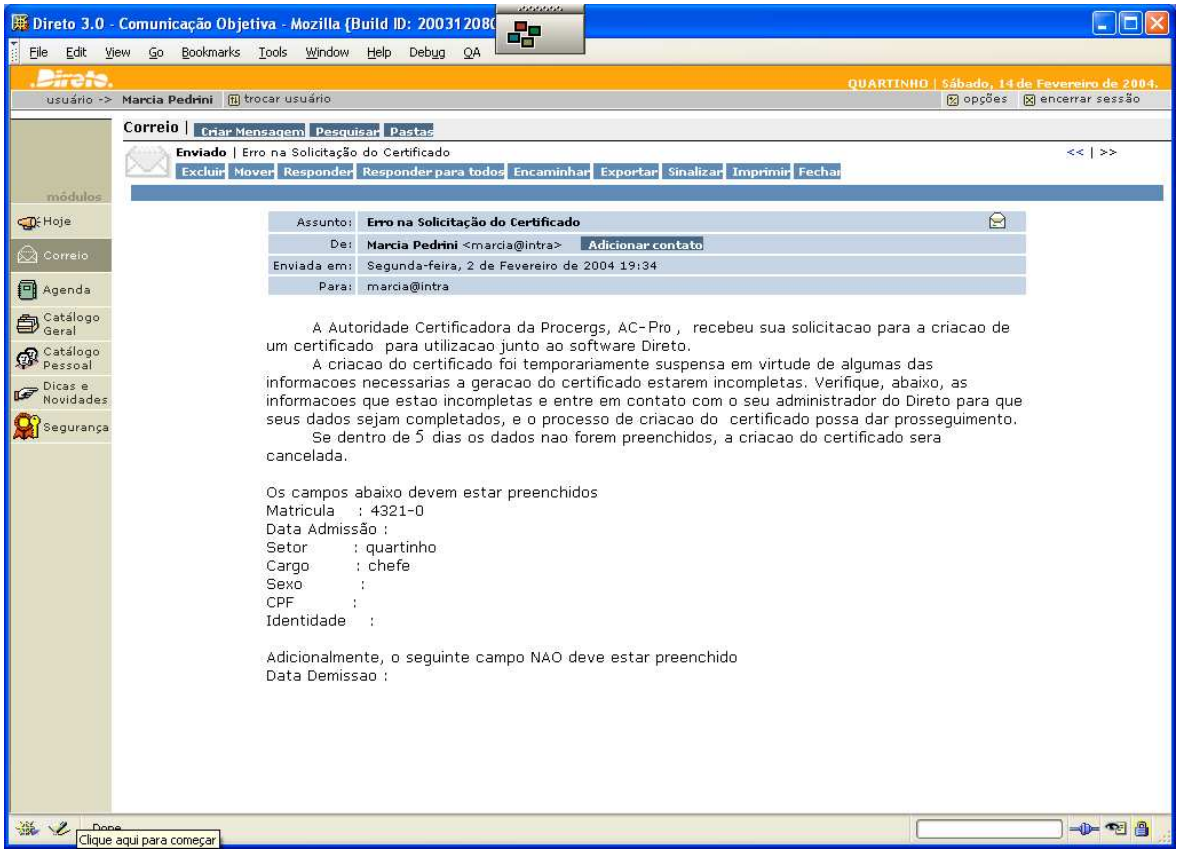

FIGURA 6.15 – E-mail avisando sobre dados incompletos

Se todos os dados estão preenchidos e são consistentes, a resposta é válida e o sistema continua com o processo de geração de certificado. A próxima etapa para a geração é a verificação se o usuário já possui um par de chaves criado. Para esta verificação, é consultado o atributo *diretoPublicKey* no LDAP. Se existe uma chave pública armazenada no LDAP, presume-se que uma chave privada está armazenada no banco PostgreSQL. A pesquisa por esta chave, então, não é necessária. O processo de geração do certificado pode prosseguir à partir deste ponto.

Se o usuário não tem uma chave pública armazenada no LDAP, é realizado o processo de geração e armazenamento do par de chaves do usuário, que utiliza um método semelhante ao descrito na figura 6.5. A diferença entre os dois métodos é que, para as chaves do usuário, o sistema irá verificar o algoritmo e tamanho de chaves no arquivo de configuração da AC-Pro.

O armazenamento do par de chaves do usuário segue o que foi definido no modelo (seção 5.5.1.4.2). A chave pública é armazenada no LDAP. Para a chave privada, deve-se proceder a cifragem da própria chave, através de um algoritmo que utiliza-se de senha. A senha, conforme definiu-se, será composta da própria senha do usuário, digitada no campo senha, quando o usuário fez seu login no sistema (conforme figura 6.14) e algumas informações cadastrais gravadas no LDAP, que são a Carteira de Identidade e o CPF. O método de criptografia da chave vai utilizar-se do padrão PKCS#5, definido no capítulo 3. O pacote JCE inclui uma implementação de criptografia com chave que é baseada no padrão PKCS#5. O método de criptografia e armazenamento da chave privada é mostrado na figura 6.16.

```
public void setPrivateKey(PrivateKey pk)
throws Exception
{
String password, senha, nome, empresa, ident, cpf;
senha = getSenha();
nome = getNome();
empresa = getEmpresa();
ident = getAtributo("identidade").trim();
cpf = getAtributo("cpf").trim();

password = senha + ident + cpf;

byte[] salt = new byte[8];

Random rng = new Random();
rng.nextBytes(salt);

byte[] cript_pk = Seguranca.encrypt(pk.getEncoded(),
password.toCharArray(), salt);

byte[] todo = new byte[cript_pk.length + 8];

for (int i=0;i<8;i++)
todo[i] = salt[i];
for (int i = 0; i < cript_pk.length; i++)
todo[i+8] = cript_pk[i];

String base64 = Base64.encode(todo);
Connection cx = null;
PreparedStatement p = null;
ResultSet rs = null;

JDBCConnectionPool pool = JDBCConnectionPool.createPool();

cx = pool.borrowConnection();
cx.setAutoCommit(false);

p = cx.prepareStatement("INSERT INTO USUARIO VALUES (?, ?, ?)");
p.setString(1, empresa);
p.setString(2, nome);
p.setString(3, base64);

int r = p.executeUpdate();

cx.commit();
p.close();
}
```

FIGURA 6.16 – Criptografia e armazenamento da chave privada

A solicitação de um certificado é enviado a AC-Pro em um formato padrão de requisição de certificados, criado pela RSA Security e chamado PKCS#10. Uma requisição de certificado [RSA2000] consiste de um nome distinto, da chave pública do requisitante e, opcionalmente, de um conjunto de atributos, e que é assinado com a chave privada da entidade requisitante do certificado. A classe utilizada para a criação da requisição do certificado pertence ao Bouncy Castle, é chamada de PKCS10CertificationRequest e tem

como argumentos o algoritmo de criptografia, o sujeito do certificado, sua chave pública e sua chave privada.

Após o recebimento da requisição do certificado, a AC_Pro faz a emissão do certificado do usuário, complementando as informações que chegaram com a requisição. Os valores que são complementados são as extensões, conforme definido no capítulo 5. Para a gravação das extensões, algumas classes já definidas pelo JDK foram utilizadas, como ExtendedKeyUsage, SubjectPublicKeyInfo, SubjectPublicKeyInfo, AuthorityKeyIdentifier, GeneralName, GeneralNames, ASN1EncodableVector, DERIA5String, DERSequence. Para a extensão CRL Distribution Point, vários passos são necessários, que incluem desde a criação de um *array* de DistributionPoint até a criação de um *array* de ReasonFlags. O certificado do usuário com suas extensões encontra-se na figura 6.17.

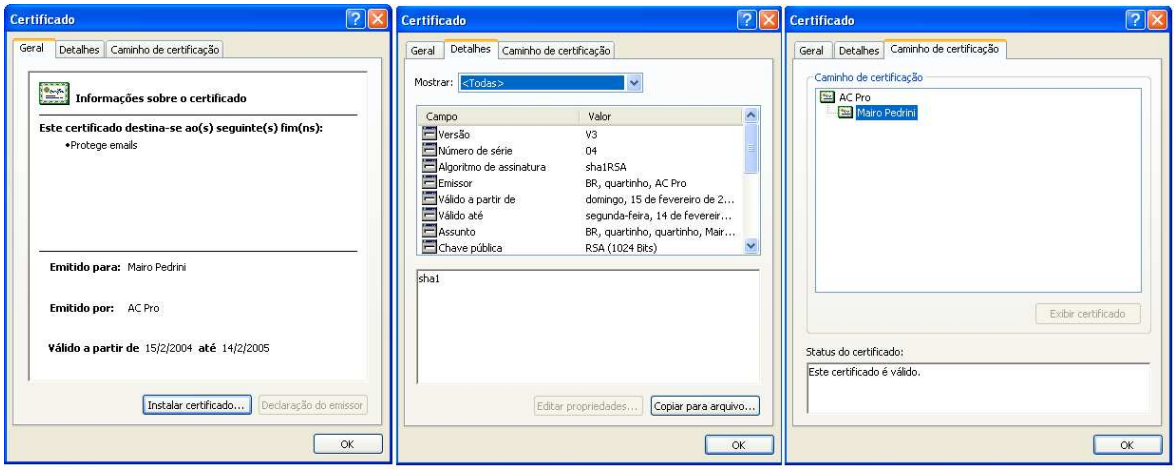

FIGURA 6.17 – Certificado do usuário

O passo seguinte é a assinatura do certificado, que consiste na criação de um resumo, através da aplicação de uma função *Hash* e da criptografia deste resumo com a chave privada da AC-Pro.

A gravação do certificado é feito na base LDAP ficando disponível para qualquer usuário que necessitar dele para assinar ou validar a autoria de uma mensagem.

Após a geração do certificado, um e-mail é enviado ao requisitante, avisando sobre a geração e disponibilização do certificado. O certificado é anexado junto a mensagem, para que o usuário possa importá-lo no seu navegador.

6.4.2 Renova Certificado

A renovação de certificados não será implementada neste protótipo. Para a renovação são seguidos os mesmos passos que para a geração de um certificado, mantendo-se o mesmo par de chaves já utilizados pelo certificado anterior.

6.4.3 Revoga Certificado

Uma LCR é uma estrutura de dados assinada contendo uma lista dos números de certificados que foram revogados, o motivo da revogação de cada um deles e a data/hora em que eles foram revogados. A AC-Pro deve emitir e divulgar periodicamente sua LCR, contendo ou não novas revogações, a fim de permitir que as partes verificadoras possam adquirir a lista completa dos certificados revogados, tendo como base as datas de divulgação de cada uma das partes das listas. O prazo de divulgação das LCRs foi definido pela DPC-Pro (Anexo A).

Para o desenvolvimento do protótipo optou-se em gravar um motivo único de revogação que é “comprometimento de chave”. Este motivo exige que, além de revogado o certificado, o usuário deve ter também suas chaves – pública e privada – revogadas. A ação de revogação do certificado, para o protótipo, executa os passos abaixo:

- remove a chave pública e o certificado do usuário da base LDAP, onde eles estão armazenados;

- remove a chave privada do usuário do banco de dados *diretodb*.

Com a finalidade de facilitar o desenvolvimento da revogação no protótipo, a gravação dos certificados revogados está sendo realizada em um arquivo gravado diretamente em disco, no mesmo diretório onde estão armazenadas as configurações do Direto. As informações gravadas neste arquivo são: o número de série do certificado e a data e hora da revogação. Na data da divulgação da LCR este arquivo é lido, cada uma das suas informações é enviada para um gerador do pacote Bouncy Castle, chamado de x509v2generator. A chamada do gerador é feita tantas vezes quantas forem os certificados gravados no arquivo. As informações da LCR são a data de emissão, data da próxima emissão, o DN da Procergs e a assinatura da AC-Pro, que é feita utilizando a sua chave privada. A lista de certificados gerada pelo Direto pode ser vista na figura 6.18.

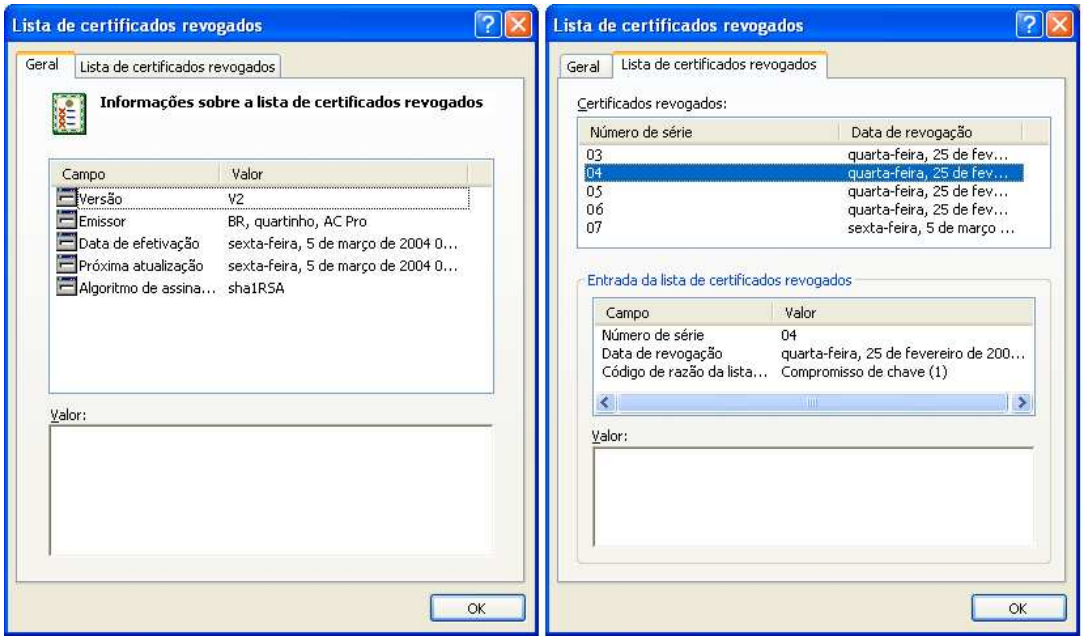

FIGURA 6.18 – LCR da AC-Pro

O próximo passo, não implementado no protótipo, seria a divulgação desta LCR para o site http://direto.org.br/LCR/AC-PRO.crl, que permitiria a um usuário acessá-la e importá-la em seu browser realizando a verificação da revogação (ou não) de determinado certificado.

6.5 Envio de e-mail

Após a criação das chaves e do certificado do usuário, o mesmo já está apto a enviar mensagens de forma segura. A interface de envio de e-mail já existente no Direto foi alterada, incluindo-se duas novas funcionalidades, Assinatura e Criptografia, que podem ser usadas separadas ou juntas, para a segurança total da mensagem. A figura 6.19 mostra a interface alterada do Direto com as novas funcionalidades.

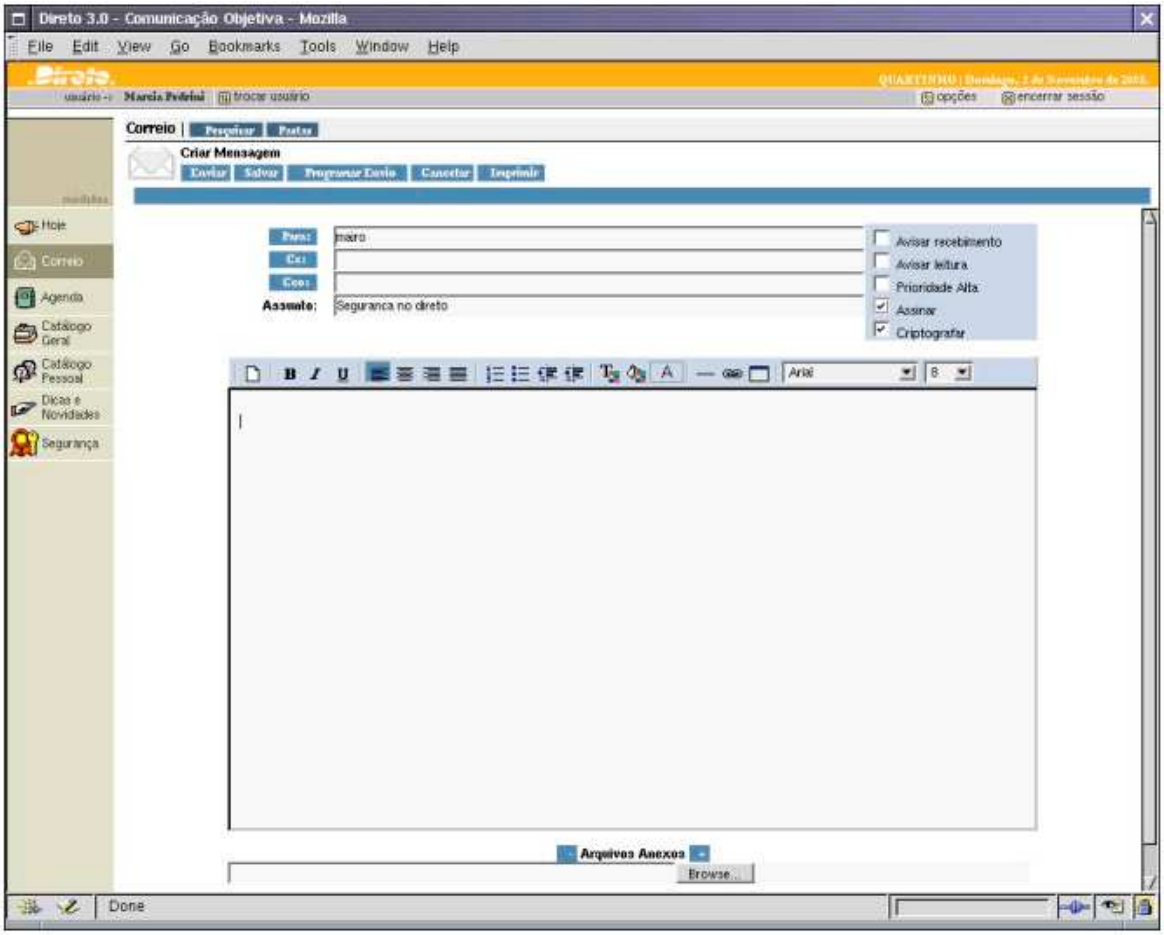

FIGURA 6.19 – novas funcionalidades no envio de e-mail

Os passos seguidos para a criptografia de uma mensagem foram:

- buscar a chave pública do destinatário da mensagem: a chave pública do destinatário se encontra na base LDAP. Para a busca desta chave foi utilizado o e-mail do destinatário. Se o destinatário não possui uma chave pública criada, uma mensagem de erro é apresentada ao usuário. O usuário tem a opção de desmarcar a opção de Criptografar, enviando o texto puro ao destinatário, ou desistir do envio, aguardando que o destinatário crie uma chave pública;

- selecionar o corpo da mensagem, digitada pelo usuário: o corpo do texto da mensagem é o texto que será cifrado;

- aplicar o algoritmo de criptografia, que foi definido na configuração da AC-Pro (sessão 1.4.4). Para o uso da classe de criptografia esbarrou-se em um erro, pouco documentado. Ao tentar-se cifrar uma mensagem, o Java retornava com a mensagem '*Unsupported keysize or algorithm parameters*'. Após muita alteração, testes de código e muita consulta a Internet, descobriu-se que o pacote JDK tem restrições, por causa da política de importação de alguns países, para uma criptografia forte. Para a liberação de uma criptografia sem qualquer limite é necessário baixar uma licença, que está disponível em

http://java.sun.com/products/jce/index-14.html. A classe de Bouncy Castle SMIMEEnvelopedGenerator se encarrega da criação de uma chave de sessão e da utilização da chave pública do receptor para a criptografia da chave de sessão.

- enviar a mensagem tendo como corpo da mensagem o novo texto, cifrado com a chave de sessão.

Os passos seguidos para a assinatura de uma mensagem consistiu em:

- buscar a chave privada no banco: a chave privada do usuário está armazenada cifrada com senha, no banco de dados PostgreSQL, e precisa ser recuperada. A busca ao banco se dá utilizando-se a empresa e o usuários logados no Direto;

- decifrar a chave privada: a chave privada foi cifrada utilizando-se uma senha composta pela senha do usuário no sistema Direto, o número da Carteira de Identidade e o número do CPF. Os mesmos argumentos serão utilizados para decifrar a chave, que também utilizou-se de um *salt*. Para decifrar foi utilizado o método PBEWithMD5AndDES do Bouncy Castle;

- assinar a mensagem: para a assinatura é utilizada uma classe, do Bouncy Castle, chamado SMIMESignedGenerator que se encarrega da assinatura da mensagem, utilizando-se da chave privada do usuário.

- a mensagem é enviada ao usuário juntamente com o certificado do usuário e o certificado da AC-Pro, para futura validação;

Para a assinatura e criptografia de uma mensagem, os passos seguidos podem ser descritos por:

- seguir os mesmos passos que foram utilizados na criptografia da mensagem, menos o parte do envio;

- após a criptografia, utilizar os mesmos passos usados para a assinatura da mensagem;

6.6 Recepção de e-mail

A tarefa de receber um e-mail também teve que ser alterada, para o correto funcionamento do correio eletrônico, fazendo uso das novas funcionalidades de criptografia e assinatura digital. Quando o usuário receber uma mensagem assinada, criptografada ou assinada e criptografada, ela será sinalizada com ícones diferentes. A interface de recepção de mensagens pelo Direto, com as novas funcionalidades está apresentada na figura 6.20.

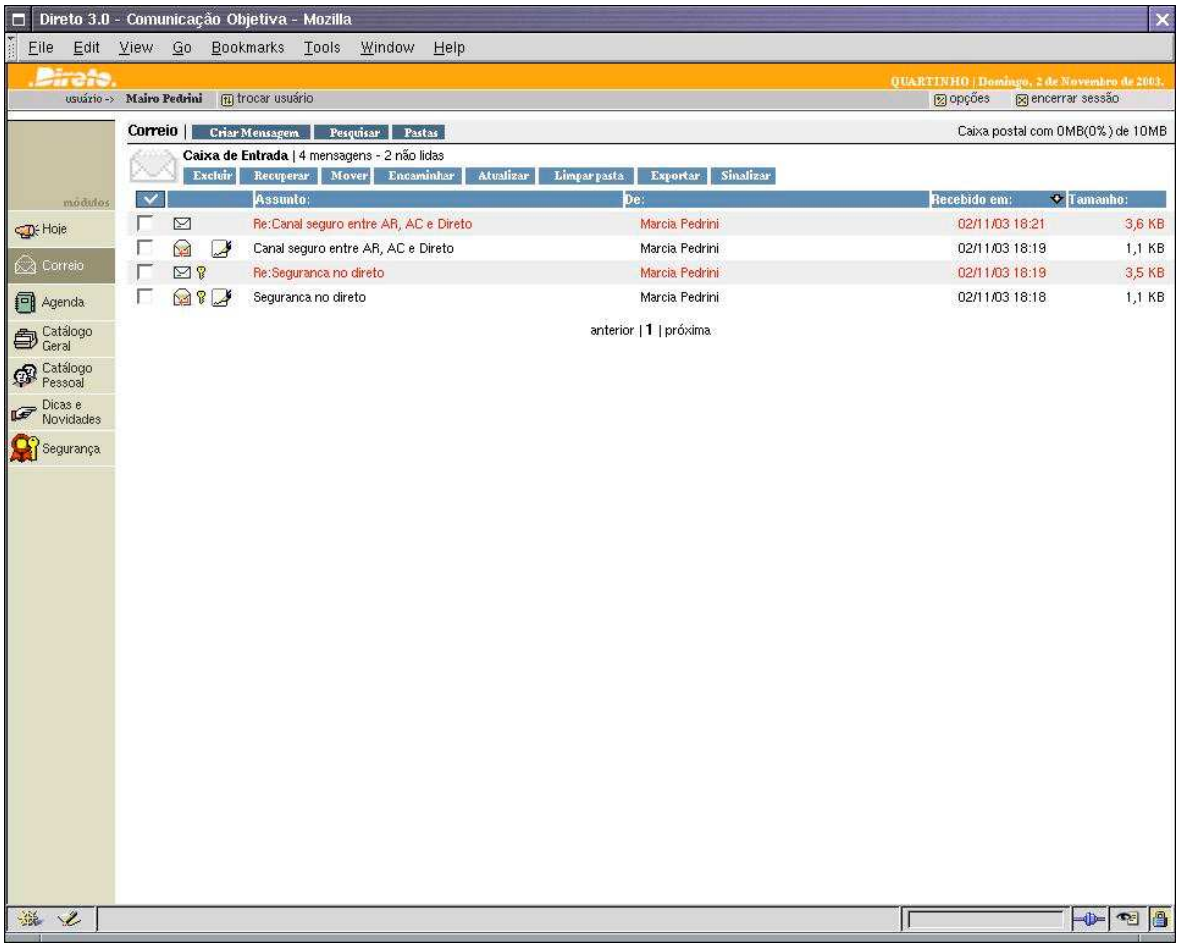

FIGURA 6.20 – Interface de recebimento de e-mail do Direto

Quando o usuário abrir uma mensagem assinada, o sistema se encarrega de seguir os passos abaixo, para validar a assinatura do remetente, atestando sua integridade.

- buscar a chave pública do remetente: a chave pública do remetente se encontra no certificado do remetente, que foi enviado junto com a mensagem;

- é utilizada uma classe de Bouncy Castle que faz a verificação da assinatura. Para o protótipo, está se levando em conta que são válidas as mensagens que foram assinadas com um certificado assinado pela AC-Pro. Se o método falha, a mensagem não tem sua autenticidade garantida, e ela é aberta conforme figura 6.21;

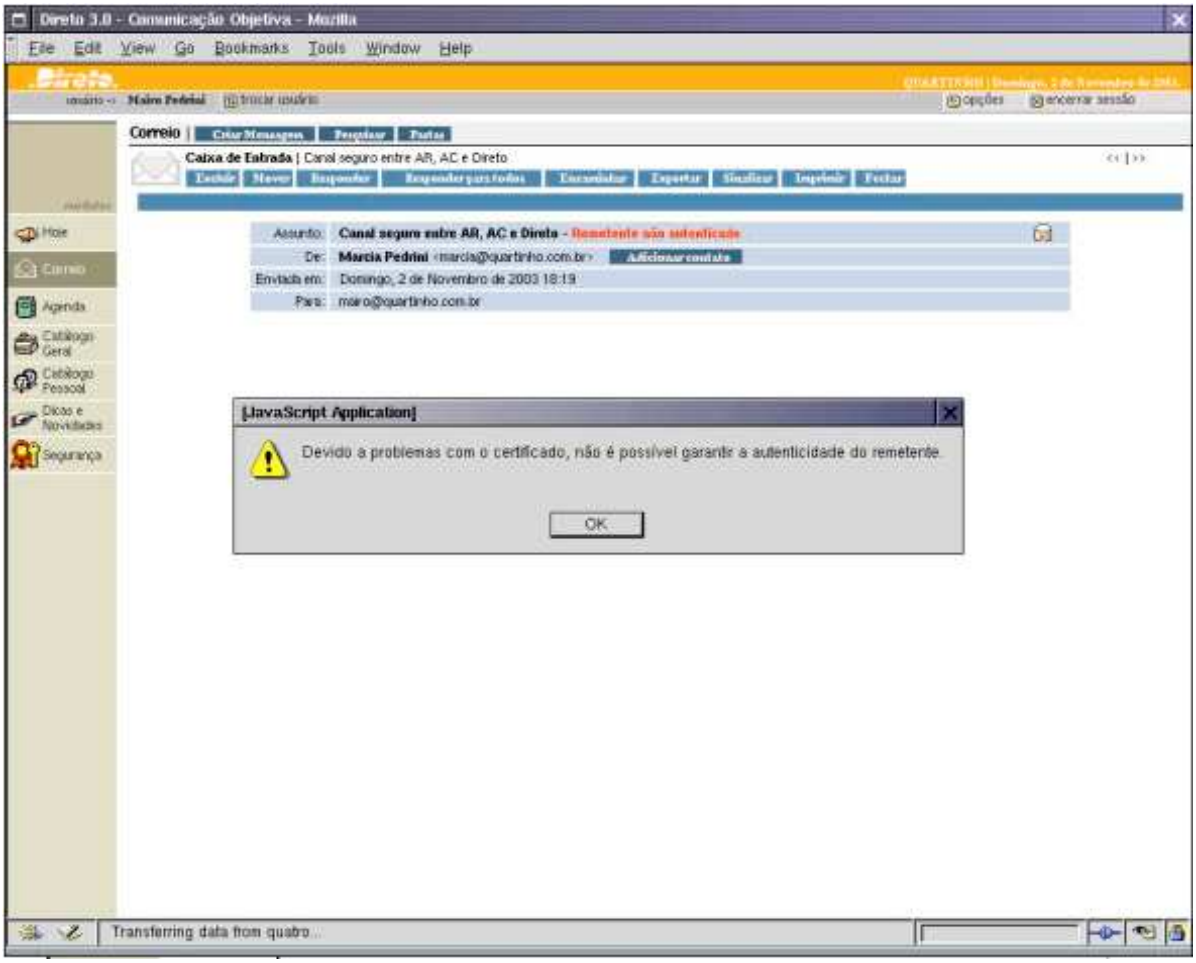

FIGURA 6.21 – Remetente não autenticado

Quando o usuário abrir uma mensagem cifrada, o sistema se encarrega de seguir os passos abaixo, para decifrar a mensagem:

- buscar a chave privada no banco: a chave privada do usuário está armazenada cifrada com senha, no banco de dados PostgreSQL, e precisa ser recuperada;

- decifrar a chave privada: a chave privada foi criptografada utilizando-se uma senha composta pela senha do usuário no sistema Direto, o número da Carteira de Identidade e o número do CPF. Os mesmos argumentos serão utilizados para decifrar a chave, que também utilizou-se de um *salt*. Para decifrara chave foi utilizado o método PBEWithMD5AndDES do Bouncy Castle;

- a mensagem é decifrada. Se a mensagem foi interceptada e alterada, decifrá-la trará um texto inconsistente, e a mensagem não poderá ser lida.

Quando o usuário abrir a mensagem cifrada e assinada, o sistema se encarregará de seguir os passos abaixo, dcifrando e atestando a autenticidade da mensagem:

- seguir os passos utilizados para a abertura de uma mensagem assinada;
- seguir os passos utilizados para a abertura de uma mensagem cifrada;

6.7 Considerações Finais

Este capítulo descreveu a implementação do protótipo da infra-estrutura de chaves públicas para o ambiente do Direto e tentou ser o mais aprofundado possível, detalhando os métodos e classes utilizadas e apresentando os erros, quando eles apareciam.

O protótipo implementado não abrangeu todo o modelo proposto, pois a tarefa de desenvolver toda uma infra-estrutura é extensa e demorada. Porém, as principais etapas do modelo foram cobertas no protótipo e isto garantiu o funcionamento do modelo proposto.

Após a conclusão do mesmo, o próximo capítulo apresentará a avaliação do protótipo, a fim de que o modelo proposto e o protótipo construído pudessem ser validados.

7 Teste e validação do protótipo

Com o objetivo de testar e validar o trabalho, foi utilizado o protótipo da infra-estrutura de chaves públicas, ICP-Pro, descrito no capítulo anterior. A validação do projeto limita-se ao escopo definido no capítulo 5, que fala sobre o modelo. A segurança referente aos outros sistemas e ferramentas utilizados, como por exemplo o LDAP e o banco de dados PostgreSQL, não fazem parte deste projeto e não foram levadas em conta. Os testes desenvolvidos são descritos nas próximas seções.

7.1 Geração de Certificado

O primeiro teste desenvolvido no protótipo refere-se à geração do certificado do usuário. Para esta geração, foi definido que o usuário necessita ter alguns campos preenchidos em sua base LDAP, que são as informações que atestam a legitimidade de uma pessoa. Estes campos são: matricula, dataadmissao, setor, cargo, datademissao (esta data deve estar nula), sexo, cpf e identidade.

Como alguns destes campos não possuem interface para cadastramento no sistema Direto, a primeira solicitação de criação de certificados apresentou erro, conforme foi demonstrado na figura 6.15.

Para a correção do problema, foi dada entrada manual nestas informações e foi novamente solicitada a geração de um certificado. A geração foi realizada com sucesso.

7.2 Criptografia de mensagem

Quando um usuário solicita a criptografia de uma mensagem, o sistema procura pela chave pública do usuário que é o destinatário da mensagem. A mensagem é cifrada com a chave pública e o destinatário vai decifrá-la com a chave privada.

Se a chave pública não for encontrada para a tarefa de criptografia, não é possível realizar a tarefa, porque o usuário não conseguirá decifrar a mensagem.

Neste caso, a tarefa é cancelada. O usuário deve tomar a iniciativa de mandar a mensagem sem criptografia, avisar o destinatário para proceder a criação de seu certificado (e consequentemente do seu par de chaves), ou desistir do envio mensagem.

Outro teste realizado com uma mensagem cifrada foi a edição e alteração de seu conteúdo diretamente na pasta do usuário. A mensagem, mesmo alterada, pôde ser aberta, porém o conteúdo apresentado estava ilegível. Este teste veio a demonstrar que uma mensagem cifrada que tenha sido interceptada e alterada por um espião é facilmente detectada por seu receptor.

7.3 Cópia de e-mail

No envio de uma mensagem cifrada foi detectada uma falha, que não havia sido prevista no modelo, sendo detectada no decorrer do protótipo. Quando envia-se uma mensagem, há a facilidade de gravar uma cópia da mensagem, junto ao remetente. A cifragem da mensagem não inclui a cifragem da cópia da mensagem.

Embora a mensagem tenha sido enviada ao destinatário de forma segura, uma cópia da mesma mensagem estaria disponível em disco para qualquer um que quisesse lê-la. A proposta encontrada, mas não desenvolvida, é não permitir que um e-mail com proposta de criptografia tenha a opção de ter uma cópia salva junto ao remetente.

7.4 Assinatura da Mensagem

Para a assinatura da mensagem, foram feitos alguns testes básicos. Todos eles resultaram em erros e a mensagem apresentada pode ser vista na figura 6.20. Os testes realizados foram:

- data / hora de validação: foi alterado o relógio do computador para uma data além da data de validade do certificado. A mensagem assinada apresentou erro porque o certificado do usuário já estava expirado;

- emissor do certificado: foi alterado o emissor do certificado, diretamente no LDAP. A abertura da mensagem apresentou erro porque a única AC válida, para o protótipo apresentado é a AC-Pro, da Procergs.

- alteração da mensagem: foi alterado o conteúdo da mensagem, diretamente na pasta do usuário. A abertura da mensagem apresentou erro porque o sistema não conseguiu validar o resumo da mensagem, através do algoritmo *hash*.

7.5 Considerações Finais

Ao longo deste capítulo foram descritos os testes principais a qual as mensagens trafegadas pelo Direto foram submetidas, após o desenvolvimentos das opções de assinatura e criptografia de mensagens. Os testes mostraram que, para a total segurança das mensagens mais algumas modificações no Direto deverão ser desenvolvidas, porém, o modelo proposto está de acordo com as necessidades apuradas e o protótipo desenvolvido conseguiu alcançar o seu objetivo.

8 Trabalhos Futuros

O modelo proposto teve a finalidade de proporcionar segurança às mensagens trafegadas pelo Direto, utilizando, para isso, o mínimo exigido de uma ICP. A partir do protótipo desenvolvido, existem outras contribuições que podem ser acrescentadas. Estas contribuições foram divididas em duas categorias: curto prazo e médio prazo.

Este capítulo descreve os trabalhos futuros com um nível de detalhamento suficiente para que possam ser realizados por demais pesquisadores que possam vir a se interessar pela tarefa.

8.1 Trabalhos futuros – curto prazo

Esta seção propõe algumas contribuições que podem ser acrescentadas ao protótipo com a finalidade de torná-lo ainda mais robusto e completo.

8.1.1 Gerenciamento de múltiplos pares de chaves

Conforme descrito no capítulo 5, o tempo de validade do par de chaves depende do propósito para a qual as chaves são geradas. Este mesmo propósito de geração dita as políticas pelas quais elas são geradas e protegidas.

O protótipo implementado faz uso de apenas um par de chaves e um único certificado para cada usuário. Porém, é comum que usuários tenham mais de um par de chaves e certificados, cada um deles servindo a um propósito distinto.

Chaves que permitem não-repúdio (assinatura) devem ser tratadas com mais cuidado que aquelas utilizadas apenas para privacidade (criptografia) das informações. Uma chave privada utilizada para fornecer assinatura digital requer um armazenamento seguro por toda a sua vida útil. Se, durante sua vida útil, ela for extraviada, é necessário apenas criar um novo par de chaves. Depois que sua vida útil terminar, ela não deve ser arquivada, mas destruída de forma segura.

Já as chaves que permitem privacidade devem ser seguramente arquivadas, mesmo depois de expirada a sua vida útil, para mais tarde poderem fornecer a decifragem dos dados legados cifrados.

O gerenciamento de múltiplos pares de chaves para cada usuário requer que uma nova alteração na estrutura do LDAP seja realizada, estendendo novamente seus dados, prevendo uma segunda chave pública e privada (ou mais, se assim for julgado necessário), e mais um certificado (ou mais, dependendo do número de par de chaves criadas) .

Um mecanismo de histórico de par de chaves pode ser desenvolvido, como um mecanismo eficiente para o arquivamento de chaves e certificados para uma posterior utilização. Este histórico deve estar seguramente armazenado e deve permitir acesso seguro

(através de um túnel de dados SSL), no caso de um usuário necessitar fazer uso desta informação. Para seu acesso, o usuário deverá ser seguramente identificado e um *log* de registro destes acessos deverá ser gerado, para atender a possíveis auditorias.

A política de certificados deve retratar a forma de uso, a validade, a forma de armazenamento e a forma de recuperação de cada um dos tipos de certificados ou par de chaves disponíveis.

8.1.2 Interface para solicitação de certificados

Conforme descrito na seção 5.5.1.7, o modelo gerado prevê a criação de certificados apenas para usuários já cadastrados no Direto. Assim, os dados necessários para a solicitação dos certificados já estão presentes no LDAP, não sendo necessária nenhuma interface especial para a captura dos dados pessoais dos usuários.

A AC-Pro segue todas as recomendações e padrões necessários para operar, tornando a PROCERGS uma Autoridade Certificadora confiável e conhecida não só por seus usuários, mas pelo público em geral. Em vista disso, ela pode começar a fornecer certificados a outros usuários interessados em ter no seu certificado a assinatura de confiança da PROCERGS, da mesma forma como fazem hoje outras Autoridades conhecidas, como a Verisign, a Certisign, o SERPRO e mesmo a ICP-Brasil.

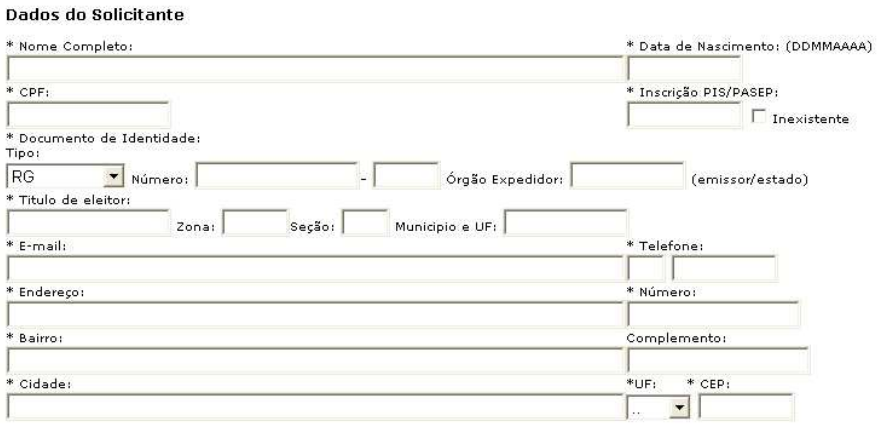

FIGURA 8.1 – Um modelo de interface

Para isso, será necessário alterar a ICP-Pro, conforme os itens abaixo:

- definir novas regras na política de certificados;
- alterar o LDAP, prevendo a utilização de novos atributos hoje não existentes, caso se torne necessário;

- criar uma interface, disponível para acesso via navegador, onde qualquer usuário possa solicitar a criação de um certificado. A figura 8.1 mostra um modelo de interface que poderia ser criado para a solicitação por parte de pessoas físicas. Outra interface semelhante seria necessária para a solicitação de certificados de pessoas jurídicas.

8.1.3 Cadeia de certificados

Uma cadeia de certificados [BUR2002] é o método mais comum utilizado para verificar a associação entre uma entidade e sua chave pública, conforme demonstra a figura 8.2. Uma parte verificadora deve verificar três coisas relacionadas a cada certificado até alcançar a raiz confiável:

- verificar se cada certificado na cadeia está assinado com a chave pública do próximo certificado;

- assegurar que cada certificado da cadeia não tenha expirado e não tenha sido revogado;

- verificar se cada certificado da cadeia está em conformidade com um conjunto de critérios definidos pelos certificados superiores da cadeia.

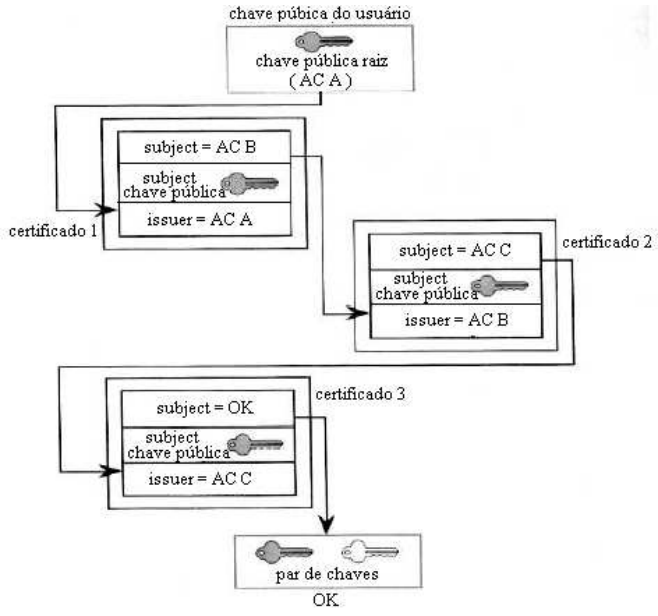

FIGURA 8.2 – Cadeia de certificados

O protótipo implementado não faz a verificação dos certificados em uma cadeia, já que todos os certificados gerados estão sendo assinados pela própria AC-Pro. Porém, o modelo proposto sugere a criação de outras ACs e de ARs. Quando estas entidades forem

criadas será necessário incorporar ao projeto um algoritmo de validação da cadeia de certificados, conforme definido pela RFC2459.

8.1.4 Revogação do certificado de uma das ACs

Assim como os usuários finais, as próprias AC são identificadas por certificados. E, assim como acontece com os certificados de usuários, também os certificados das ACs podem ser revogados, suas chaves privadas podem ser comprometidas ou o prazo de validade dos certificados pode expirar. Nestes casos, deve existir um tratamento especial a ser realizado que deve estar detalhado e deve ser divulgado através da DPC-Pro.

Os passos a serem seguidos quando da revogação de uma das ACs subordinadas podem ser descritos por:

- a AC-Pro é informada por comunicações seguras sobre a revogação do certificado de uma das ACs subordinadas;
- a revogação é processada e uma nova LCR é publicada imediatamente;
- são notificados os titulares e usuários de certificado.
- a AC-Pro revoga os certificados de AR antigos;
- um novo par de chaves é gerado e armazenado, e é realizada a emissão de um novo certificado de AC;
- procedimentos de *backup* de chaves privadas são executados;
- cada AR submete um pedido para um certificado novo, que é validado pela AC-Pro e resulta na geração de novos pares de chaves e certificados para as ARs;
- certificado de AR novo é entregue à AR junto com a novo certificado da AC;
- cada AR identifica todos os certificados ativos emitidos pela AC comprometida sob sua responsabilidade e submete um pedido de revogação para cada um deles;
- a AC-PRO processa a revogação e publica uma LCR imediatamente (por razões de desempenho a LCR é publicada somente após um certo volume de revogações);
- a emissão de novos certificados para cada usuário é feita de forma automática, sendo os usuários avisados que um novo certificado foi gerado.

Processos semelhantes, porém com algumas variações, devem ser seguidos para o casos de revogação de certificado de alguma das ARs ou mesmo pela extinção da AC-Pro.

8.2 Trabalhos futuros – médio prazo

Embora o protótipo implementado seja funcional, os avanços na área de segurança não param de crescer. Por isso, algumas contribuições são sugeridas de forma a manter a ICP-Pro em conformidade com as novas tecnologias e avanços que surgem nesta área.

8.2.1 Divulgação de certificados revogados

O método mais comum [ANK2002] de publicação das revogações é uma LCR. A LCR é simplesmente uma lista assinada dos certificados que foram revogados. Vários outros mecanismos de publicação estão disponíveis na literatura, porém, não existe um consenso sobre qual das formas é a melhor. O estudo realizado neste trabalho mostrou que, em todas as técnicas existentes, a geração de uma LCR completa, com todos os certificados revogados sempre é utilizada. Por isso, esse foi o método empregado.

Como o método de publicação de uma LCR completa não é a melhor, por causa do seu crescimento com o passar do tempo, outro mecanismo provavelmente será implementado. Um outro mecanismo estudado e que está se tornando bastante utilizado, já tendo um padrão definido, é o OCSP (Online Certificate Status Protocol), que foi projetado para prever o *status* de um certificado em tempo real, usando um servidor que fica respondendo às solicitações sobre cada certificado.

Um estudo mais aprofundado sobre o OCSP e os outros métodos existentes deve ser realizado, de forma a definir e implementar uma forma mais leve e rápida de disponibilizar as informações de revogação dos certificados.

8.2.2 Mobilidade

A tecnologia *wireless*, como os telefones digitais e PDAs (Personal Digital Assistants), estão se tornando rapidamente populares, e os usuários finais já são capazes de utilizar estes equipamentos para acessar muitas aplicações de m-commerce. Neste cenário, pode-se facilmente vislumbrar um usuário do Direto querendo acessar seus e-mails ou mesmo sua agenda de compromissos utilizando-se desta tecnologia.

O acesso a Internet através de *wireless* tem algumas limitações em termos de tamanho de banda, capacidade de armazenamento, recursos de memória, tempo de vida da bateria e interface do usuário. Mas uma coisa continua a mesma: neste ambiente, segurança também é fundamental.

Para o mundo do wireless está sendo desenvolvida [MUN2002] uma adaptação da infra-estrutura de chaves públicas, chamado WPKI (Wireless-PKI). WPKI é usado em um ambiente WAP (Wireless Application Protocol) para gerenciar certificados, para servidores e clientes, usando WTLS (Wireless Transport Layer Security).

O uso da nova tecnologia WPKI vai depender não apenas de alterações nas funcionalidades já desenvolvidas para a segurança do correio eletrônico, mas também da alteração do sistema, para que ele possa operar em um dispositivo móvel.

8.2.3 Armazenamento de chaves por dispositivos de hardware

Conforme descrito na seção 2.2.6, vários são os mecanismos existentes para se proteger a chave privada de um usuário. A opção de armazená-la através de criptografia com a utilização de senha não é a melhor delas, já que alguém que tenha acesso a local onde ela está armazenada e que conheça o método de geração de senha utilizado, poderia vir a utilizá-la. Porém, esta é a única solução que pode ser utilizada sem nenhum tipo de investimento adicional, utilizando-se apenas software.

Um estudo sobre os diversos mecanismos conhecidos, a forma como eles funcionam, os benefícios e falhas conhecidas de cada um, o método de recuperação da chave e um estudo sobre custo *versus* benefício serão importantes. O Direto deve disponibilizar, além do método já implementado, a liberdade do usuário escolher como deseja manter a privacidade de sua chave.

As rotinas que utilizam a chave privada para criptografar e/ou assinar as mensagens devem ser alteradas, já que a chave poderá estar armazenada de diferentes formas, em diferentes locais.

8.2.4 Registro de Data / Hora

Uma assinatura digital garante a autenticidade de quem a assina. Porém, para garantir que a mesma mensagem que hoje é autêntica não seja, no futuro, repudiada, é necessário que haja algum controle de data / hora em que o documento foi assinado para garantir que a revogação de seu certificado se deu após esta data / hora.

A literatura cita várias formas de datação de documentos: método linear, método de ligação e método randômico [MAT2000] [BAY1992]. A RFC3161 [ADA2001] define um formato de uma requisição enviada para uma TSA (Autoridade de Carimbo de Tempo) e a resposta, devidamente assinada, que ela retorna.

Para que as mensagens assinadas que trafegam pelo Direto consigam garantir o não-repúdio de sua assinatura, será necessário desenvolver algum tipo de recurso de datação.

9 Conclusão

O rápido crescimento da Internet seguidas pelo aumento constante da troca de informações de forma eletrônica, tem proporcionado também a rápida evolução de um fator negativo para a segurança eletrônica: o aumento das tentativas de ataque por pessoas interessadas em obter ou alterar informações sigilosas e até mesmo em se fazer passar por outra pessoa para obter algum ganho.

Quando se fala em comprometimento de uma informação através de uma falha de segurança, nem sempre se tem uma idéia muito clara do que isso significa, podendo parecer para leigos ou iniciantes da área um exagero tanto investimento em segurança. Porém, podemos exemplificar a motivação do desenvolvimento deste trabalho, para o Direto, por alguns motivos básicos para manter as mensagens seguras:

- vazamento da informação: informações como relatórios financeiros, informações do setor de Recursos Humanos, informações sobre licitações, sistemas, programas ou projetos em desenvolvimento podem vir a parar nas mãos de pessoas de fora da organização;

- perda da reputação: se alguma informação vazar, através de um ataque bem sucedido a um servidor de e-mail do Direto, os usuários perderão a confiança no sistema e poderão ter receio em utilizá-lo. Todos os anos de estudo e implementação utilizados no projeto podem ser colocados em risco por causa de uma brecha na segurança;

- perdas financeiras: espiões (que podem ser pessoas de fora da organização, funcionários ou mesmo ex-funcionários) que consigam acesso a informações sigilosas trafegando via e-mail referente a projetos que envolvam dinheiro, poderão se utilizar destas informações para tirar vantagem para si próprio ou eventualmente vendendo informações para as partes interessadas.

Um dos pontos mais importantes quando se fala em segurança não foi abordado no decorrer deste trabalho, mas é importante que seja levado em conta: são as pessoas envolvidas no processo. De nada adiantam ferramentas seguras se não houver uma conscientização prévia da finalidade de seu uso. Não haverá serventia o modelo descrito no decorrer desta dissertação se os usuários não julgarem necessário utilizarem-se dos certificados com o propósito de tornar as suas informações seguras. E, principalmente, de nada terá valido todo o esforço deste trabalho se os usuários não forem treinados e conscientizados que suas informações são importantes e confidenciais não devendo serem mantidas suscetíveis a interceptação por desconhecidos, espiões ou mesmo bisbilhoteiros.

Com a evolução da Internet até acordos com poder jurídico estão saindo do papel e sendo realizados por meios eletrônicos. A distância não é mais problema para obter, por exemplo, a assinatura de presidentes ou diretores de vários órgãos estatais, localizados a quilômetros de distância, para a realização de uma grande transação. Entretanto, é necessário que estes documentos tenham validade jurídica.

Em 24 de Agosto de 2001 o governo instituiu [PIN2001] [BRA2001], pela Medida Provisória n.º 2.200, a Infra-Estrutura de Chaves Públicas Brasileira (ICP-Brasil), que os documentos emitidos eletronicamente passam a ter validade jurídica. Segundo o artigo 10 desta MP, são legais os documentos em forma eletrônica produzidos com a utilização de processo de certificação disponibilizado pela ICP-Brasil, ou os que utilizem certificados não emitidos pela ICP-Brasil, desde que admitido pelas partes como válido ou aceito pela pessoa a quem for oposto o documento.

A dissertação aqui descrita consistiu de um modelo e no desenvolvimento de um protótipo para a criação de uma infra-estrutura de chaves públicas para a utilização junto ao e-mail do Direto. O objetivo inicial do trabalho era o estudo da melhor alternativa já disponível para uso com ferramentas de software livre e que tivessem por finalidade prover os quatro quesitos básicos de segurança que são: autenticação, integridade, não-repúdio e privacidade. O estudo das alternativas mostrou que esta tecnologia ainda está em evolução. Muitas das ferramentas encontradas estão incompletas, trazem erros ou não são totalmente confiáveis.

A solução encontrada, então, foi desenvolver uma infra-estrutura de chaves públicas completa, com suas entidades, o relacionamento entre estas entidades e as políticas de segurança necessárias a manter a segurança dos dados trafegados através do Direto. Para o desenvolvimentos desta proposta algumas dificuldades foram sendo encontradas e tiveram que ser vencidas:

- a documentação em relação a alguns passos da certificação são confusos e não trazem uma explicação do seu completo funcionamento, como é o caso da revogação dos certificados, que não possui um consenso sobre qual das formas existentes é a melhor;

- o entendimento e uso dos padrões se tornou essencial para criar uma interoperabilidade, já que acredita-se na evolução constante deste modelo, permitindo no futuro, que ele faça trocas de certificados com outras instituições.

Um protótipo foi desenvolvido e demonstrou que o modelo desenvolvido abrange todas as necessidades para o tráfego e armazenamento seguro da mensagens, bem como a validação dos usuários (emissores e receptores) das mensagens.

Testes realizados demonstraram que os principais objetivos da interceptação de mensagens por inimigos foram tratados, desde que os métodos disponibilizados na aplicação sejam devidamente utilizados.

Referências

[ADA2001] ADAMS, C. et al. **Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP).** 2001. Disponível em: <http://www.ietf.org/rfc/rfc3161.txt>. Acesso em: ago. 2002.

[AME1996] AMERICAN BAR ASSOCIATION. **Digital Signature Guidelines.** 1996. Disponível em: <http://www.abanet.org/scitech/ec/isc/dsg.pdf>. Acesso em: ago. 2002.

[ANK2000] ANKNEY, R. C.; **Certificate Revocation Mechanisms**, [S.l.]: CertCo, Inc., Disponível em: <http://cnscenter.future.co.kr/resource/rsc-center/vendor-wp/certco/revoc.pdf>. Acesso em: 2000.

[APA2001] APACHE SOFTWARE FOUNDATION. **Apache HTTPD Project.** Disponível em: <http://httpd.apache.org>. Acesso em: 15 nov. 2001.

[ARC1999] ARCHITECTURE for Public-Key Infrastructure (APKI). 1999. Disponível em: <http://www.opengroup.org/onlinepubs/009219899/toc.pdf>. Acesso em: ago. 2002.

[ART2003] ARTICSOFT. **Introduction to Public Key Infrastructure**. 2003. Disponível em: <http://www.articsoft.com/wp_pki_intro.htm>. Acesso em: ago. 2003.

[BAL2002] BALINSKI, R. **Filtragem de Informações no Ambiente do Direto.** 2002. Disponível em: <www.inf.ufrgs.br/procpar/direto/trabalhos/dissertacao-pdf.PDF>. Acesso em: ago. 2002.

[BAY1992] BAYERY, D.; HABER, S.; STORNETTA, W. S. **Improving the Eficiency and Reliability of Digital Time-Stamping**. 1992. Disponível em: <http://www.surety.com/docs/bhspap.pdf>. Acesso em: ago. 2002.

[BRA2001] BRASIL. Medida Provisória nº 2.202, de 24 de agosto de 2001. Institui a Infra-Estrutura de Chaves Públicas Brasileira – ICP - Brasil, transforma o Instituto Nacional de Tecnologia da Informação em autarquia, e dá outras providências. Disponível em: <http://www.planalto.gov.br/ccivil_03/MPV/2200-2.htm>. Acesso em: Jan. 2004.

[BUR2002] BURNETT, S.; PAINE, S. **Criptografia e Segurança:** o guia oficial RSA. Rio de Janeiro: Campus, 2002.

[CAR2000] CARVALHO, D. B. **Segurança de Dados com Criptografia, Métodos e Algoritmos**. [S.l.]: Book Express, 2000. 218p.

[CHO1999] CHOKHANI, S.; FORD, W. **Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework**. 1999. Disponível em: <http://www.ietf.org/rfc/rfc2527.txt>. Acesso em: ago. 2002.

[CON1998] CONN, D. **Gatekeeper** : A Strategy for Public Key Technology Use in the Government. 1998. Disponível em: <http://www.noie.gov.au/publications/GatekeeperStrategy.pdf>. Acesso em: ago. 2002.

[COV2002] COVELL, C; BELL, M. **OpenCA Guides for 0.9.2+**. Disponível em: <http://www.openca.org/openca/docs/files/openca-guide.pdf>. Acesso em: ago. 2002.

[DAE1999] DAEMEN, J.; RIJMEN V. **AES Proposal:** Rijndael. 1999. Disponível em: http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael-ammended.pdf. Acesso em: ago. 2002.

[DAG2000] DAGEFORDE, M. **The Java tutorial**: trail - security in Java 2 SDK 1.2. Disponível em: <http://web2.java.sun.com/docs/books/tutorial/security1.2/index.html>.Acesso em: mar. 2000.

[DON2002] DONLEY, C. **LDAP Programming, Management and Integration**. [S.l.]: Manning Publications, 2002.

[FAP2000] FAPERGS. **Projeto Sistemas Avançados para Comunicação Eletrônica** : software aberto de correio, agenda e catálogo: edital 06/2000. Disponível em: <http://www.inf.ufrgs.br/procpar/direto/ProjetoFapergsEdital062000.html>. Acesso em: ago. 2002.

[FAR1998] FARLEY, J. **Java Distributed Computing**. [S.l]: O´Reilly, 1998.

[FER2001] FERREIRA, J. B. C.; GOLDANI, C. A. **Modelos de Políticas de Certificados**. UniCERT. 2001. Disponível em: <http://www.unicert.com.br/files/doc/unicert12.doc>. Acesso em: ago. 2002.

[FER2003] FERRO, W. R. **Comércio Eletrônico e a Segurança da Rede**: uma visão tecnológica. 2003. Disponível em: <http://www.ead.fea.usp.br/Semead/6semead/MQI.htm>. Acesso em: ago. 2003.

[FEW2002] FEWER, S. **SSL**: A discussion of the Secure Socket Layer. 2002. Disponível em: < http://www.securitytechnet.com/security/application.html>. Acesso em: ago. 2002.

[FRE1996] FREIER, A. O.; KARLTON, P.; KOCHER, P. C. **The SSl Protocol Version 3.0**. 1996. Internet Draft. Disponível em: <http://wp.netscape.com/eng/ssl3/ssl-toc.html>. Acesso em: ago. 2002.

[FRE2003] FREUND, F. F., FREUND, G. P.; IGNACZAC, L. **Uma visão sobre Infra-estrutura de Chaves Públicas**. 2003. Disponível em: <http://www.ctai.senai.br/artigos/revista20031/Rev.%203ed_Art10.PDF>. Acesso em: ago. 2003.

[GAR2001] GARMS, J.; SOMERFIELD, D. **Professional Java Security**. [S.l.]: Wrox Press, 2001.

[GUN2000] GUNTER, C. A; JIM, T. Generalized Certificate Revocation. In: ANNUAL SYMPOSIUM ON PRINCIPLES OF PROGRAMMING LANGUAGES. 2000. Boston. **Proceedings...** New York: ACM Press, 2000. p. 316-329.

[HOU1999] HOUSLEY, R. et al. **Internet X.509 Public Key Infrastructure Certificate and CRL Profile**. 1999. Disponível em: <http://www.ietf.org/rfc/rfc2459.txt>. Acesso em: ago. 2002.

[IBM1999] IBM SECURITY. **Public Key Infraestructure**: The PKIX Reference Implementation Project (aka Jonah). 1999. Disponível em: <www.di.uminho.pt/~mac/9900/ca/projecto/wp_pkix.pdf>. Acesso em: ago. 2002.

[ILI2000] ILIADIS, J. et al. **Evaluating Certificate Status Information Mechanisms.** 2000. Disponivel em: <http://www.spinellis.gr/pubs/conf/2000-CCS-CSI-Eval/html/csi-eval.pdf>. Acesso em: ago. 2002.

[JOH2000] JOHNER, H. et al. **Deploying a Public Key Infrastructure**. Austin: IBM. International Technical Support Organization. 2000. Disponível em: <http://www.redbooks.ibm.com/redbooks/pdfs/sg245512.pdf>. Acesso em: ago. 2002.

[JOS2000] JOSANG, A.; PEDERSEN, I. G.; POVEY, D. **PKI Seeks a Trusting Relationship**. Disponível em: <dstc.edu.au/papers/pkitrust.ps>. Acesso em: ago. 2000.

[KAR1996] KARLTON, P.; KOCHER, P. C. **The SSL Protocol Version 3.0**. 1996. Disponível em: <http://wp.netscape.com/eng/ssl3/draft302.txt>. Acesso em: ago. 2002.

[KLE2001] KLENSIN, J. **Simple Mail Transfer Protocol**. 2001. Disponível em: <http://www.ietf.org/rfc/rfc2821.txt?number=2821>. Acesso em: ago. 2002.

[KNU1998] KNUDSEN, J. **Java Cryptography**. O´Reilly, 1998.

[KUH2001] KUHN, D. et al. **Introduction to Public Key Technology and the Federal PKI Infrastructure**. [S.l.]: National Institute of Standards and Technology.

2001. Disponível em <http://csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf>. Acesso em: ago. 2001.

[LIA2002] LIAU, C. Y.; BRESSAN S.; TAN K. **Efficient Certificate Revocation** : A P2P Approach. Singapore: Departament of Computer Science, National University of Singapore. Disponível em: <http://xena1.ddns.comp.nus.edu.sg/p2p/effcertr.pdf>. Acesso em: dez. 2002.

[MAR2001] MARTINS, A. **Autoridade Certificadora para Acesso Seguro**. [S.l.]: Laboratório RAVEL / COPPE / UFRJ. 2001. Disponível em: <http://www.lockabit.coppe.ufrj.br/downloads/academicos/CA.pdf>. Acesso em: ago. 2002.

[MAT2000] MATSUURA, K.; IMAI, H.; **Digital timestamps for dispute settlement in electronic commerce**: generation, verification, and renewal. Disponível em: <http://imailab-www.iis.u-tokyo.ac.jp/Members/kanta/iceis02cr.pdf>. Acesso em: dez. 2000.

[MUN2002] MUÑOZ-TAPIA, J. L.; FORNÉ-MUÑOZ, J. **CPC-OCSP**: an Adaptation of OCSP for m-Commerce. **The European Online Magazine for the IT Professional**. 2002. Disponível em: <http://www.upgrade-cepis.org/issues/2002/6/up3-6Munoz.pdf>. Acesso em: ago. 2003.

[MYE1999] MYERS, M. et al. **X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP**. 1999. Disponível em: <http://www.ietf.org/rfc/rfc2560.txt>. Acesso em: ago. 2002.

[OPE2003] THE OPENLDAP FOUNDATION. **OpenLDAP 2.1 Administrator's Guide**. 2003. Disponível em: <www.openldap.org/doc/admin/>. Acesso em: ago. 2003.

[OPE2004] OPENSSL PROJECT. **Miscellaneous OpenSSL Document**. Disponível em: <http://www.openssl.org/>. Acesso em jan. 2004.

[PER1999] PERLMAN, R. An Overview of PKI Trust Models. **IEEE Network** .New York, v.13, n.6, p.38-43, Nov. / Dec. 1999.

[PIN2001] PINHEIRO, P. Documentos eletrônicos serão reconhecidos. **O Estado de São Paulo**, São Paulo, 11 jul. 2001. Disponível em: <http://www.mct.gov.br/mct%20site/internet/english/sobre/namidia/ctnamidia/11_07.htm>. Acesso em: ago. 2002.

[PRO2000] PROJETO Sistemas Avançados para Comunicação Eletrônica – Software Aberto de Correio, Agenda e Catálogo. 2000. Disponível em: <http://www.inf.ufrgs.br/procpar/direto/>. Acesso em: 29 maio 2001.

[REG2001] REGULY, A. **OpenSSL RauTu**. 2001. Disponível em: <http://reguly.net/alvaro/linux/security/openssl/rautu/OpenSSL_RauTu/>. Acesso em: ago. 2002.

[RSA2000] RSA DATASECURITY. **PKCS#10 v1.7**: Certification Request Syntax Standard. 2000. Disponível em: <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-10>. Acesso em: ago. 2002.

[RSA1993] RSA DATASECURITY. **PKCS#8:** Private-Key Information Syntax Standard. 1993. Disponível em: <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-8>. Acesso em: ago. 2002.

[SOM2000] SOMMERER, A. **The Java tutorial**: trail - JAR files. Disponível em: <http://web2.java.sun.com/docs/books/tutorial/jar/index.html>. Acesso em ago. 2002.

[SUN2001] SUN MICROSYSTEMS. **JavaServer Pages**. Disponível em: <http://java.sun.com/products/jsp>. Acesso em: 11 nov. 2001.

[SUN2002] SUN MICROSYSTEMS. **Java 2 SDK tools.** Disponível em: <www.java.sun.com/products/jdk/1.2/docs/tooldocs/tools.html>. Acesso em: mar. 2002.

[SUN2003] SUN MICROSYSTEMS. **Java Secure Socket Extension (JSEE)**. Disponível em <http://java.sun.com/products/jsse/>. Acesso em: mar. 2003.

[SUN2004] SUN MICROSYSTEMS. **Tutorial Keytool – Key and Certificate Management Tool**. Disponível em: <http://java.sun.com/products/jdk/1.2/docs/tooldocs/solaris/keytool.html>. Acesso em: jan. 2004

[THE2001] THE MOD_SSL package. Disponível em: <http://www.modssl.org/:. Acesso em: 15 nov. 2001.

[THO1997] THOMAS, M. D. et al. **Programando em Java para a Internet**. São Paulo. Makron Books, 1997.

[WAH1997] WAHL, M. et al. **Lightweight Directory Access Protocol (v3)**: Attribute Syntax Definitions. 1997. Disponível em: <http://www.ietf.org/rfc/rfc2252.txt>. Acesso em: ago. 2002.

[WEB1995] WEBER, R. F. Criptografia Contemporânea. In: SIMPÓSIO DE COMPUTADORES TOLERANTES A FALHAS, 6., 1995, Canela. **Anais...** Porto Alegre: Instituto de Informática da UFRGS, 1995. P. 7-32.

[WEI2001] WEISE, Joel. **Public Key Infrastructure Overview**. [S.l.]: Sun Microsystems 2001. Disponível em:

<http://www.sun.com/solutions/blueprints/0801/publickey.pdf>. Acesso em: ago. 2002.

[XEN1999] XENITELLIS, S. **The Open-Source PKI Book**: A Guide to PKIs and Open-Source Implementations. 1999. Disponivel em: <http://ospkibook.sourceforge.net>. Acesso em: ago. 2002.

[XML2001] XML: Extensible Markup Language. Disponível em: <http://www.w3.org/TR/REC-xml> Acesso em: 11 nov. 2001.

[XSL2001] XSL Transformations (XSLT). Disponível em: <http://www.w3.org/TR/xslt>. Acesso em: 11 nov. 2001.

[ZUR1999] ZURKO, M. E. et al. Jonah: Experience Implementing PKIX reference freeware. In: USENIX SECURITY SYMPOSIUM, 8., 1999, Washington. **Proceedings...** Berkeley, CA: USENIX Association, 1999. Disponível em: < http://www.usenix.org/publications/library/proceedings/sec99/full_papers/zurko/zurko.pdf>.Acesso em: ago. 2002.

Anexo Declaração de Práticas de Certificação da Autoridade Certificadora da Procergs (DPC da AC-Pro)

1. Introdução
Esta DPC descreve as práticas que a AC-Pro utiliza ao emitir e administrar certificados.

1.1 Visão Geral
A AC-Pro emite certificados com as seguintes finalidades:
1) Autenticação pessoal
2) Assinaturas digitais
3) Criptografia
Esta DPC provê informações que descrevem:
1) As práticas empregadas pela AC-Pro para fornecer serviços de certificação;
2) O uso de tecnologias e processos auxiliares para fornecer a estrutura operacional subjacente.

1.2 Autoridades Certificadoras
O propósito primário da AC-Pro é prover serviços de certificação aos Titulares de Certificado dentro dos seus respectivos domínios de Política de Certificados (PC). A AC-Pro sempre que implementar um novo tipo de certificado deverá publicar a Política de Certificados (PC) e a Declaração de Práticas de Certificação da Autoridade Certificadora da Procergs associada ao certificado.

1.3 Autoridades de Registro
A AR vinculada da AC-Pro tem a responsabilidade de tratar solicitações de certificado, autenticando a identidade do candidato, aprovando ou rejeitando então a solicitação.

1.4 Titulares de Certificados
Titulares de Certificados são as entidades – pessoas físicas, autorizados pela AR responsável a receber um certificado digital emitido pela AC-Pro, para sua própria utilização.

2 Disposições Gerais
2.1 Obrigações e Direitos
Nos itens a seguir são descritas as obrigações gerais da AC-Pro.

2.1.1 Obrigações da AC-Pro
A AC-Pro cumpre as seguintes obrigações:
1. Gerar e gerenciar o seu par de chaves criptográficas;
2. Assegurar a proteção de sua chave privada;
3. Assegurar a proteção da chave privada do usuários;
4. Notificar os seus usuários quando ocorrer suspeita de comprometimento de sua chave, emissão de novo par de chaves e correspondente certificado, ou o encerramento de suas atividades;
5. Distribuir o seu próprio certificado;

6. Emitir, expedir e distribuir os certificados de AR vinculadas e de Titulares de Certificados;
7. Informar a emissão do certificado ao respectivo solicitante;
8. Revogar, quando necessário, os certificados por ela emitidos;
9. Emitir, gerenciar e publicar suas listas de Certificados Revogados (LCR);
10. Publicar em sua página *Web* esta DPC;
11. Investigar comprometimento e suspeitas de comprometimento de sua chave privada;

2.1.2 Obrigações das AR

As AR vinculadas à AC-Pro, cumprem as seguintes obrigações:

1. Operar de acordo com esta DPC e a PC a qual se vincula;
2. Receber solicitações de emissão ou de revogação de certificados;
3. Confirmar a identidade do solicitante e a validade da solicitação;
4. Encaminhar a solicitação de emissão ou de revogação de certificados;
5. Informar aos respectivos titulares a emissão ou a revogação de seus certificados;
6. Disponibilizar os certificados emitidos pela AC-Pro aos seus respectivos solicitantes;
7. Identificar e registrar todas as ações executadas;

2.1.3 Obrigações de Titulares de Certificados

Titulares de Certificados sob esta DPC cumprem as seguintes obrigações:

1. Fornecer, de modo completo e preciso, todas as informações necessárias para sua identificação;
2. Utilizar os seus certificados e chaves privadas de modo apropriado, conforme o previsto na PC correspondente;
3. Conhecer os seus direitos e obrigações, contemplados pela DPC e pela PC correspondente;
4. Informar à AC-Pro, através de sua AR, qualquer comprometimento de sua chave privada e solicitar a imediata revogação do certificado correspondente;
5. Assinar o Termo de Responsabilidade do certificado;
6. Notificar imediatamente a AR de qualquer erro ou defeito nos certificados, ou de qualquer mudança subseqüente na informação do certificado;
7. Utilizar os pares de chaves e certificados conforme PC implementada pela AC-Pro;

2.1.4 Direitos do Usuário de Certificado (Terceira Parte Confiável)

Considera-se Usuário de Certificado, uma entidade que confia no teor, validade e aplicabilidade do certificado digital. Constituem direitos do Usuário de Certificado:

1. Verificar, a qualquer tempo, a validade do certificado. Um certificado emitido pela AC-Pro é considerado válido quando:

• não constar da LCR da AC emitente;
• não estiver expirado; e
• puder ser verificado com o uso de certificado válido da AC-Pro.

2. Recusar a utilização do certificado para fins diversos dos previstos na PC correspondente.

O não exercício desses direitos não afasta a responsabilidade da AC responsável e do Titular do Certificado.

2.1.5 Obrigações do Repositório
O repositório da LCR é disponibilizado na página *Web* http://direto.org.br/LCR/AC-PRO.crl. O mesmo está disponível para consulta durante 24 (vinte e quatro) horas por dia, 7 (sete) dias por semana.

2.2 Publicação de informação da AC-Pro
A AC-Pro mantém página *Web* https://direto.org.br/AC-PRO , que contém as seguintes informações:
1) Esta DPC;
2) A PC que implementa;
3) Certificado da AC-Pro;

2.3 Freqüência de publicação
São publicadas sempre as versões mais recentemente aprovadas desta DPC prontamente após a aprovação.
A publicação das LCR será disponibilizada conforme definido na PC que a AC-Pro implementa.
Certificados são publicados prontamente conforme sua geração e emissão nos casos em que as PC assim o definirem.

2.4 Sigilo
A chave privada de assinatura digital da AC-Pro será gerada e mantida pela própria AC-Pro, que é responsável pelo seu sigilo.
A AC-Pro será responsável pela geração, manutenção e garantia de sigilo da chave privada dos titulares de certificados de assinatura digital emitidos pela AC-Pro, bem como pela divulgação ou utilização indevidas dessas mesmas chaves.

3 Identificação e Autenticação
3.1 Registro Inicial
Neste item e nos seguintes, a DPC descreve os requisitos e os procedimentos gerais utilizados pelas AR vinculadas à AC-Pro no processo inicial de identificação dos solicitantes de certificado. Os requisitos e procedimentos específicos estarão detalhados nas PC implementadas pela AC-Pro..

3.1.1 Tipos de nomes
No domínio da AC-Pro, o atributo sujeito nos certificados emitidos para Titulares de Certificado , são do tipo *Distinguished Name*, contendo sempre o seu nome no formato previsto pelo padrão ITU X.500.

3.1.2 Método para comprovar a posse da Chave Privada
A mensagem de solicitação de certificado obedece ao formato PKCS#10.

3.2 Geração de novo par de chaves antes da expiração do atual.
Os Titulares de Certificado serão comunicados da necessidade da renovação de acordo com o prazo configurado junto a AC-Pro, se assim for configurado pela AC-Pro, ou terão seus certificados renovados automaticamente.

3.3 Geração de novo par de chaves após revogação
Para o caso específico de revogação de um certificado de titular pela AC-Pro, após a revogação de seu certificado o titular do certificado deverá executar os processos regulares de geração de seu novo par de chaves.

3.4 Solicitação de Revogação
Os procedimentos utilizados para confirmação da identidade de uma entidade solicitante de revogação do certificado dessa entidade são sempre documentadas pela AR-Pro.

4 Requisitos Operacionais
4.1 Solicitação de Certificado
Os requisitos e procedimentos mínimos necessários para a aceitação de uma solicitação de emissão de certificado devem ser:
1. A comprovação de atributos de identificação constantes do certificado;
2. A aceitação, pela parte do solicitante, do termo de aceitação da Ac-Pro.

4.2 Emissão de Certificado
As AR devem tomar os cuidados necessários ao aceitar e processar solicitações de certificado. Elas devem obedecer às práticas descritas nesta DPC e a qualquer exigência imposta pela PC implementada pela AC-Pro.
Um certificado será considerado válido a partir do momento de sua emissão.

4.3 Aceitação de Certificados
O recebimento de um certificado pelo Titular de Certificado e o uso subseqüente das chaves e certificado, constitui aceitação do certificado por parte do Titular de Certificado.
Aceitando um certificado, o Titular de Certificado:
1. Concorda estar de acordo com as responsabilidades continuas, obrigações e deveres impostas a ele pelo Termo de Responsabilidade e PC implementada pela AC-Pro e esta DPC;
2. Garante que por seu conhecimento, nenhuma pessoa sem autorização teve acesso à chave privada associada com o certificado;
3. Afirma que as informações de certificado fornecidas durante o processo de solicitação são verdadeiras e foram publicadas dentro do certificado com precisão.

4.4 *Suspensão e Revogação de Certificados*
4.4.1 Circunstâncias para revogação
A revogação pode ser descrita como a permanente inutilização de um certificado. As circunstâncias para revogação são especificadas na PC aplicável. Um certificado de AR ou de Titular de Certificado, é revogado tipicamente quando:
1. As chaves foram comprometidas ou há suspeita de comprometimento por:
• roubo, perda, revelação ou modificação da chave privada;
2. Existe abuso deliberado de chaves e certificados, ou uma desobediência significativa de exigências operacionais contidas no Termo de Responsabilidade, PC associada ou das práticas desta DPC;
3. Um Titular de Certificado deixa a comunidade de interesses da PC sob a qual foi emitido, por exemplo:
• um Titular de Certificado organizacional deixa o emprego ;

• uma AR cessa sua operação;
• ocorre o falecimento de um Titular de Certificado;
4. Há uma emissão imprópria ou defeituosa de um certificado devido a:
• um pré-requisito para a emissão do certificado que não foi satisfeito;
• descoberta de uma evidência objetiva no certificado que leva a acreditar como sendo falso o certificado;
• erro na entrada de dados ou outros erros de processamento;
5. Uma informação do certificado torna-se inexata, por exemplo quando o Titular de Certificado muda o nome;
6. Um pedido corretamente formatado é recebido do Órgão empregador do Titular do Certificado;
7. Um pedido validado é recebido de um terceiro autorizado, por exemplo:
• uma determinação judicial;
8. Certificado de uma AR é revogado após comprovação da má utilização deste certificado;

4.4.2 Quem pode solicitar revogação

Revogações de certificado podem ser iniciadas tipicamente por:
1) Solicitação do Titular do Certificado;
2) Solicitação do gerente do setor ao qual o titular é vinculado;
3) Pela AC-Pro;
4) Por uma AR vinculada;

4.4.3 Procedimento para solicitação de revogação

As práticas envolvidas no processo de um pedido de revogação variam, dependendo da identidade do solicitante, e devem ser detalhadas nas PC implementadas pela AC-Pro.
Como diretrizes gerais, fica estabelecido que:
1. solicitante da revogação de um certificado será identificado;
2. As solicitações de revogação, bem como as ações delas decorrentes serão registradas e armazenadas;
3. As justificativas para a revogação de um certificado serão documentadas;
4. processo de revogação de um certificado terminará com a geração e a publicação de uma LCR que contenha o certificado revogado.

4.4.4 Circunstâncias para suspensão

Um certificado poderá ser suspenso quando o usuário entrar em férias.

4.4.5 Quem pode solicitar suspensão

Os solicitantes de uma suspensão de certificado são os mesmos que podem solicitar a revogação de um certificado;

4.4.6 Procedimento para solicitação de suspensão

Os procedimentos para a suspensão de um certificado são os mesmos que para a revogação de um certificado;

4.4.7 Freqüência de emissão de LCR

A LCR da AC-Pro será divulgada 1 (uma) vez por semana. Os números de série de certificados de qualquer entidade final que estejam revogados aparecem na LCR emitida pela AC-Pro. Estes números permanecem nas LCR emitidas até a data de expiração dos certificados ser atingida, sendo removidos na primeira LCR emitida após data de suas

expirações. As LCR são emitidas mesmo que não haja nenhuma mudança ou atualização, para assegurar a periodicidade da informação.

4.4.8 Requisitos para verificação de LCR

Todos os certificados revogados no domínio da AC-Pro são listados na LCR que pode ser acessada no endereço *Web* contido no próprio certificado.

Antes de aceitar um certificado os Titulares de Certificados devem verificar a situação do mesmo na LCR corrente.

4.4.9 Requisitos especiais para o caso de comprometimento de chave

Quando houver comprometimento ou suspeita de comprometimento da chave privada, o Titular do Certificado deverá comunicar imediatamente a AC-Pro. Os requisitos específicos aplicáveis a revogação do certificado nestas circunstâncias são descritos nas PC implementadas pela AC-Pro.

Os meios utilizados para comunicar um comprometimento ou suspeita de comprometimento de chave são descritos nas PC implementadas pela AC-Pro.

4.5 Arquivamento de Registros

4.5.1 Tipos de registros arquivados

As seguintes informações são registradas e arquivadas pela AC-Pro e AR:

1) solicitações de certificados;
2) solicitações de revogação de certificados;
3) notificações de comprometimento de chaves privadas;
4) emissões e revogações de certificados;
5) emissões de LCR;

4.6 Troca de chave

Os certificados emitidos pela AC-Pro, e as respectivas chaves criptográficas geradas por seus Titulares, possuem prazos de validade que inicia a partir do momento da geração do certificado. Expirado este prazo, um novo par de chaves e um novo certificado deve ser gerado. Os prazos dos certificados emitidos pela AC-Pro estão detalhados nas PC aplicáveis. As AR da AC-Pro se encarregam de avisar aos Titulares de Certificados antes da expiração dos seus certificados para que o processo de solicitação de novo certificado não cause impacto aos mesmos, ou inicia o processo automaticamente, se assim configurado.

5 Controles de Segurança Física, Procedimental e de Pessoas

6 Controles Técnicos de Segurança

6.1 Geração e Instalação do Par de chaves

6.1.1 Geração do Par de Chaves

Pares de chaves são gerados somente pelo Titular do Certificado correspondente. Os procedimentos específicos estão descritos em cada PC implementada.

6.1.2 Entrega da chave privada à entidade titular

Item não se aplica.

O Titular do Certificado é responsável pela geração e a guarda da sua chave privada.

6.1.3 Entrega da chave pública para emissor de certificado
Chaves públicas são entregues ao emissor de certificado por meio de uma troca *on-line* utilizando funções automáticas do software de certificação da AC-Pro.

6.1.4 Disponibilização de chave pública da AC-PRO para usuários
A forma para a disponibilização do certificado da AC-Pro será:
2) Diretório;
3) Página *Web* da AC-Pro;

6.1.5 Tamanhos de chave
O tamanho das chaves criptográficas do certificado da AC-Pro é de 2048 (dois mil e quarenta e oito) *bits.*

6.1.6 Propósitos de uso de chave (conforme campo "Key usage" na X.509 v3)
A chave privada da AC-PRO é utilizada para a assinatura dos certificados por ela emitidos e de suas LCR.
As chaves privadas dos Titulares de Certificados emitidos pela AC-PRO podem ser utilizadas para Assinatura Digital ou Sigilo. Certificados de assinatura serão utilizados em aplicações como confirmação de identidade no correio eletrônico.

6.2 Outros Aspectos do Gerenciamento do Par de Chaves
6.2.1 Períodos de uso para as chaves pública e privada
A chave privada da AC-Pro e chaves privadas de assinatura digital de Titulares de Certificados por elas emitidos são utilizadas apenas durante o período de validade dos certificados correspondentes. As correspondentes chaves públicas podem ser utilizadas durante todo o período de tempo determinado pela legislação aplicável, para verificação de assinaturas geradas durante o prazo de validade dos respectivos certificados.

7 Perfis de Certificado e LCR
7.1 Perfil do Certificado
Todos os certificados emitidos pela AC-PRO estão em conformidade com o formato definido pelo padrão ITU X.509.
7.1.1 Número(s) de versão
Todos os certificados emitidos sob a hierarquia da AC-PRO implementam a versão 3 do padrão ITU X.509, de acordo com o perfil estabelecido na RFC 2459.
7.1.2 Extensões de certificados
A AC-PRO implementa para os certificados de pessoa física as seguintes extensões:
1) "*Authority Key Identifier*": o campo **keyIdentifier** contém o resumo SHA-1 da chave pública da AC-Pro;
2) "*Key Usage*", crítica: somente os bits **digitalSignature**, **nonRepudiation, keyEncipherment e dataEncipherment** são ativados;
3) "*Certificate Policies*": contém o endereço *URL* da página *Web* da AC-Pro com a DPC da AC-Pro;
4) "*CRL Distribution Points*": contém o endereço *URL* da página *Web* onde se obtém a LCR da AC-Pro;
6) "*Subject Alternative Name*":com a sub-extensão "*rfc822Name*" contendo o endereço e-mail do titular do certificado.

7) “*Extended-key-usage*”: contendo “*E-mail protection*” (OID 1.3.6.1.5.5.7.3.4).

7.1.3 Formatos de nome

Nos certificados emitidos, o nome do titular do certificado, constante do campo “*Subject*”, adota o “*Distinguished Name”*
(DN) do padrão ITU X.500/ISO 9594, da seguinte forma:
C = BR
O = Procergs
OU = sigla do órgão de trabalho
CN = nome do titular do certificado

7.2 Perfil de LCR

7.2.1 Número (s) de versão

As LCR geradas sob a hierarquia da AC-Pro implementam a versão 2 do padrão ITU X.509, de acordo com o perfil estabelecido na RFC 2459.

7.2.2 Extensões de LCR e de suas entradas

A AC-Pro adota as seguintes extensões de LCR:
1) **“***Authority Key Identifier*”: contém o resumo SHA-1 da chave pública da AC-PRO.
2) “*CRL Number*”, não crítica: contém número seqüencial para cada LCR emitida.

8 Administração de Especificação