ESTUDO E IMPLEMENTAÇÃO DE INFRA-ESTRUTURA DE CHAVES PÚBLICAS COM APLICAÇÃO EM CONTROLE DE ACESSO A REDES SEM FIO

Alessandro Martins

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

Aprovada por:

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

Profa. Noemi de La Rocque Rodriguez, Dr.

Prof. Sérgio Barbosa Villas-Boas, Ph.D.

Prof. Sérgio Lima Netto, Ph.D.

RIO DE JANEIRO, RJ - BRASIL
MARÇO DE 2004

MARTINS, ALESSANDRO

Estudo e Implementação de Infra-estrutura de Chaves Públicas com Aplicação em Controle de Acesso a Redes sem Fio [Rio de Janeiro] 2004

VII, 178p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2004)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Infra-estrutura de Chaves Públicas
2. Criptografia
3. Redes sem Fio

I. COPPE/UFRJ II. Título (série)

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

ESTUDO E IMPLEMENTAÇÃO DE INFRA-ESTRUTURA DE CHAVES PÚBLICAS COM APLICAÇÃO EM CONTROLE DE ACESSO A REDES SEM FIO

Alessandro Martins
Março / 2004

Orientador: Luís Felipe Magalhães de Moraes
Programa: Engenharia de Sistemas e Computação

As infra-estruturas de chaves públicas (ICPs) têm sido proclamadas como a tecnologia que irá tornar o comércio e os relacionamentos usando a Internet realmente seguros em todos os aspectos. Porém ainda existem problemas para atigir este objetivo, entre eles o alto custo das soluções, a falta de conhecimento da tecnologia por parte do mercado e dos desenvolvedores de aplicações. Esses problemas junto com outros inerentes ao estado atual da tecnologia, como a questão da divulgação das informações sobre a situação dos certificados formam o cenário desta dissertação. Nela avalia-se comparativamente às opções comerciais a utilização de software livre para a implantação de ICPs e seleciona-se uma para uso. Além disso, os dois principais métodos de verificação de situação de certificados atuais, que são as LCRs e o OCSP, são analisados com relação as suas funcionalidades, desempenho e aplicabilidade. Devido à inexistência de uma implementação ***standalone*** e ***opensource*** do OCSP foi realizada uma. A ICP selecionada para uso junto com a implementação feita do OCSP foi avaliada funcionalmente em um cenário real no controle de acesso de usuários de uma rede sem fio a uma rede infra-estruturada. Os resultados mostraram que cada método de divulgação de situação de certificados possui características que se adequam melhor a ambientes específicos, por exemplo, o OCSP se mostrou uma opção mais adequada a futura geração de aplicações. Finalmente, o impacto sobre a rede causado por qualquer um dos métodos de revogação pode ser controlado de forma efetiva ajustando para tal alguns parâmetros do processo de revogação.

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

STUDY AND IMPLEMENTATION OF PUBLIC-KEY INFRASTRUCTURE WITH APPLICATION IN ACCESS CONTROL TO WIRELESS NETWORKS

Alessandro Martins
March / 2004

Advisor: Luís Felipe Magalhães de Moraes
Department: System Engineering and Computer Science

The infra-structures of Public Keys have been proclaimed as the technology that will change the commerce and internet's relationships into a complete secure way in all aspects. Although there are still problems to achieve this point, among all of them the high cost of solutions and the missing knowledge of technology by commerce people and applications developers. These problems join the inherent state of art technology problems like the matter of information broadcast about certificates state and it becomes the scenery of this dissertation. It will be comparatively evaluated the commercial free software options to Public-Key Infrastructures (PKIs) implementation and it will be chosen one to use. Also the two main verification methods of the certification's state (CRL and OCSP) will be exhaustively analyzed about their functionalities, performance and applicability. A standalone opensource OCSP's implementation was made because there was no one so far. The PKI, selected to be used with OCSP's implementation, was functionally evaluated in real access wireless network user control scenery within infra-structured network. The results show that each method for certification state situation broadcast has some qualities that allow a better adaptation to a specific environment for instance, OCSP is the best next generation application option and the impact question over a network caused by some of them may be effective controlled.

Lista de Acrônimos

AC - Autoridade Certificadora

AR - Autoridade Registradora

ASN.1 - Abstract Syntax Notation One

DER - Distinguished Encoding Rules

ICP - Infra-estrutura de Chave Pública

OID - Object Identifieres

PEM - Privacy Enhanced Mai

PKC - Public Key Certificate

PKC - Public Key Certificate

SSL - Secure Socket Layer

LCR - Lista de Certificados Revogados

OCSP - Online Certificate Status Protocol

CRL - Certificate Revocation List

PKI - Public-Key Infrastructure

Lista de Figuras

Lista de Tabelas

Índice

Capítulo

1 Introdução

O crescimento do uso da Internet[1]pode ser avaliado com base na expansão dos recursos disponíveis e usuários nos últimos anos. Em 1984 o número de hosts (servidores de recursos) girava em torno de 1.000, saltando para 100.000 em 1989, atingindo 1.000.000 em 1992 e chegando a 3.200.000 em 1994. Em 1997 existiam cerca de 10.000.000 de hosts e mais de 30 milhões de usuários. Já no início de 2001, eram mais de de 110.000.000 de hosts e mais de 400 milhões de usuários. Os últimos dados de dois grandes analisadores da Internet mostram que em maio de 2003 havia 580 milhões de usuários, segundo o NUA[1], enquanto o Telcordia NetSizer[2], em setembro de 2003 relatava mais de 840 milhões.

Esse crescimento se deve, entre outros fatores, a evolução dos meios de transmissão, a redução dos custos e a proliferação da tecnologia da informação tanto no meio comercial quanto residencial e com isso o uso de mensagens eletrônicas e do comércio eletrônico tem-se difundido a grandes taxas.

Todos os dias empresas e indivíduos usam a Internet para executar inúmeras de transações online. As empresas compartilham arquivos e informações confidenciais via email ou por redes privadas virtuais, clientes de bancos atualizam suas contas, fazem pagamentos e requisitam produtos de todas as formas e funções que são pagos via ordens eletrônicas de seus computadores pessoais, governos emitem certidões com validade legal e empresas da área médica disponibilizam atestados e pareceres para acesso dos seus clientes ou outros médicos, tudo via Internet.

Isso tem ocorrido por que a Internet, como infra-estrutura de comunicação, tem despertado grande interesse das empresas e governos por vários fatores importantes, como

1. Neste texto, **internet** se refere a coleção de redes (que podem apresentar-se isoladas) usando TCP/IP enquanto **Internet** se refere a rede global de computadores, onde convergem todas as redes não-isoladas.

redução dos custos e expansão do mercado consumidor o que tem motivado a passagem de empreendimentos da "velha economia" para este novo ambiente. O gráfico da evolução dos domínios *dotcom* fornecido pelo *Internet Software Consortium*[3] dá uma idéia deste crescimento na América Latina e no mundo.

No Brasil não existem muitas pesquisas de domínio público sobre a Internet. A mais completa disponível é fornecida pela perceria IBOPE-eRatings chamada Web Shoppers[4]. Nela é possível acompanhar a evolução de alguns indicadores ligados ao comércio eletrônico no país mostrando que o Brasil também segue a tendência mundial nessa área. Estima-se[4] que no ano de 2004 os negócios pela Internet movimentem algo em torno de 5 trilhões de dólares no mundo cabendo do Brasil menos de 1.5% dessa movimentação.

Como ferramenta de negócios e comunicação, a Internet tem o potencial de substituir outros veículos como o telefone e o fax nas relações diárias, principalmente com a adoção em larga escala das redes sem fio. Nas mãos erradas, a tecnologia da Internet também pode ser usada para interceptar e forjar mensagens, capturar informações sensíveis, bisbilhotar e defraudar organizações e indivíduos.

Esses ameaças existem porque as comunicações via Internet são, devido aos protocolos de comunicação empregados, inerentemente anônimas e públicas. Além disso, a suite mais empregada para a comunicação na Internet atualmente, o TCP/IP[2], possui diversas brechas[5][6][7], algumas específicas das redes sem fio[8], que precisam ser fechadas para que este meio possa ser efetivamente um canal apto e seguro para comunicação, livre de incidentes de segurança danosos[10].

No passado, a segurança das redes de dados era principalmente de interesse militar e acadêmico. É intuitivo pensar que quando os problemas surgiram no setor comercial e privado (a Internet atual) as abordagens anteriores, especialmente as relacionadas às redes sem fio, não foram adequadas. Essa inadequação se deve, entre outros fatores, a atual dimensão da rede e seu uso, ao avanço das técnicas de ataque com o conseqüente desenvolvimento de novas ferramentas de fácil obtenção e emprego e com o aumento da capa-

2. Nesta dissertação, a menos que explicitamente citada outra versão, sempre que mencionada a suite TCP/IP será considerada apenas versão 4.

cidade computacional, facilitando a violação de sistemas e protocolos, até mesmo daqueles que façam uso da criptografia.

Como o número de organizações que usam a Internet para fazer negócios não deixa de crescer, torna-se imprescindível cobrir as brechas e possibilitar o estabelecimento de um vínculo de confiança entre pessoas e empresas que nunca se encontraram e possivelmente nunca se encontrarão novamente, num ambiente com as características da Internet.

Uma das soluções mais promissoras para os problemas citados é o emprego de uma Infra-estrutura de Chave Pública (ICP).

Uma ICP é um sistema complexo composto em suma por três partes: softwares, hardwares e procedimentos operacionais e pode ser vista como um substrato sobre o qual são implementados mecanismos de segurança que podem ser usados por qualquer aplicação que compreenda a forma de acesso a esses recursos de segurança.

Para desempenhar seu papel uma ICP deve ser capaz de realizar todo o processo de emissão de certificados, armazenamento, publicação (ou acesso on-line), revogação e arquivamento para verificação futura. Em conseqüência disso, esse sistema constitui um artefato computacional complexo, com capacidade de comunicação, processamento e armazenamento com requisitos muito específicos.

Com uma ICP torna-se possível transpor as limitações dos mecanismos existentes no TCP/IP, garantindo assim confidencialidade a uma comunicação (quer seja por email, ou outra forma qualquer), integridade dos dados, autenticidade e não repúdio, citando apenas as principais, além de ser possível se certificar da identidade e confiar em um usuário da Internet.

Numa ICP, o objeto central é o certificado digital. Ele é emitido por uma entidade confiável chamada de Autoridade Certificadora (AC) e seu conteúdo declara uma associação entre uma chave digital e um conjunto de informações que podem ser de identificação de um indivíduo, por exemplo.

O certificado digital, quando usado para identificação, é semelhante a uma cédula de identidade na Internet. Ela contém a identificação do usuário na Internet e informações que variam com o emprego do certificado ou com a política adotada pela autoridade emissora. Essa identificação é imprescindível em diversas situações, como para controlar o

acesso de usuários e para efetivar o comércio usando a Internet. Quando necessário, a associação pode ser desfeita de forma quase instantânea.

Por isso o estudo das ICPs se destaca. Sua importância num mercado cada vez mais globalizado e sua aplicabilidade, que permeia muitas relações, como as internas de uma empresa (entre departamentos) e as externas (entre clientes e fornecedores), abre caminho para novas interações e grandes mudanças, principalmente no relacionamento entre governos (troca de informações de segurança nacional, como dados de possíveis terroristas) e entre o governo e a sociedade (declarações por exemplo).

1.1 Definição do Problema

O cenário exposto no tópico anterior pode não revelar o impacto do domínio da tecnologia empregada para tornar o comércio eletrônico via Internet possível. Como citado, toda a comunicação na Internet é inerentemente anônima e pública e isso dificulta o comércio, por isso o certificado digital surge com tanta força.

Com a inevitável expansão do uso da Internet como meio de negócio, uma grande quantidade de informações sensíveis e transações comerciais passarão a ser realizadas através dela, usando como apoio a tecnologia das infra-estruturas de chave pública.

É de se esperar que neste cenário um número muito grande de certificados exista e que estes continuem sendo emitidos a altas taxas. Como conseqüência, o problema mudará de forma. Deixará se ser um problema de identificação e passará a ser um problema de gerenciamento e verificação de identidade.

Atualmente, o processo mais comum de revogação de certificados é feito através da inclusão dos identificadores dos certificados em um tipo especial de certificado que armazena uma lista sequêncial. Esta estrutura é chamada de Lista de Certificados Revogados, ou simplesmente LCR e é gerada e divulgada pela autoridade certificadora que emitiu os certificados ou por uma outra a qual a tarefa foi delegada.

Um relatório do MITRE[9] feito com base no uso das LCRs quantificou o consumo de recursos de comunicação necessários à distribuição de informações sobre revogação e

concluiu que este consumo tem o potencial de ser o aspecto de maior custo em uma ICP com muitos usuários.

Além disso, como o volume de certificados e conseqüentemente de operações realizadas com eles pode vir a ser tão significativo, espera-se que, sem os devidos controles, estas LCRs contenham muitos certificados e com isso atinjam tamanhos significativos que irão consumir largura de banda devido também ao grande número de requisições das mesmas. Outro problema da LCRs é a frequência com que elas são emitidas, durante estes intervalos os certificados revogados passarão erroneamente como válidos.

A combinação desses dois problemas com as topologias presentes em [11] para os caminhos de validação pode gerar uma falha ou lentidão no processo e causar impactos sérios, como a impossibilidade de validar um caminho de certificação.

Estes possíveis problemas se tornam ainda mais factíveis e são a motivação desta dissertação pois: (a) as soluções existentes, devido ao impacto estratégico, são caras, estrangeiras, proprietárias e originam de um grupo extremamente pequeno de empresas desenvolvedoras; (b) o desenvolvimento de pilotos de testes e pesquisas pelo setor acadêmico não tem sido amplamente realizado[3], dada a importância da tecnologia em questão; (c) a grande maioria dos sistemas e aplicações atuais usam LCRs que não são adequadas pois não possuem a performance e as funcionalidades necessárias a demanda de verificações e novas aplicações por vir.

Outro fator que agrega mais dificuldade ao problema é que com o rápido crescimento da Internet, o mercado de tecnologia da informação não conseguiu preparar profissionais capacitados em segurança à mesma taxa, gerando uma lacuna nessa área[12]. Como conseqüência disso existe uma dificuldade na criação de novos aplicativos que façam uso desta nova tecnologia e possibitem a evolução tecnológica e comercial do país.

3. Numa pesquisa feita em 21/01/04 utilizando a ferramenta Google apenas em páginais nacionais, a palavra chave OCSP retornou apenas 111 referências, onde algumas se referem a outro acrônimo

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo desta dissertação é analisar algumas soluções para infra-estruturas de chaves públicas (ICPs) e mostrar a viabilidade do uso de software livre para a implantação de uma ICP na qual aplicativos que façam uso da certificação digital possam ser desenvolvidos e testados.

Este ambiente abrigará os componentes básicos de uma ICP, incluindo as LCRs, além da implementação de um protocolo de verificação de certificados online chamado OCSP (*Online Certificate Status Protocol*) desdobrado em duas partes: um cliente genérico e um servidor, necessários para a validação do protocolo. Esse protocolo é o representante mais maduro da sua classe, que visa resolver os possíveis problemas de desempenho e funcionalidade das LCRs.

1.2.2 Objetivos Específicos

• Introduzir o leitor aos conceitos de criptografia, assinatura eletrônica e certificação digital relacionados ao tema;

• Descrever os componentes, mecanismos e objetos necessários a uma ICP, em especial:

- as entidades de registro e certificação,
- os repositórios de certificados,
- a hierarquia de validação dos certificados,
- o formato dos certificados e das lista de certificados revogados.

• Analisar as soluções disponíveis em software livre aptas a serem empregadas no desenvolvimento de uma ICP adequada ao ambiente de testes proposto e comparar as suas características com as características das soluções comerciais;

• Apresentar os problemas de segurança e de desempenho do método de revogação de certificados baseado em listas de certificados e propor alternativas adequadas para as redes sem fio.

• Implementar e avaliar o protocolo de verificação de certificados OCSP com

relação a dois aspectos: sobrecarga de processamento no cliente e servidor e consumo de recursos de comunicação na troca de mensagens.

• Integrar a infra-estrutura de chaves públicas resultante ao sistema AirStrike[13] de autenticação de usuários em uma rede de sem fio e avaliar o resultado com relação a sua adequação e desempenho para uma possível utilização prática.

1.3 Resultados e Contribuições

Esta dissertação aborda uma tecnologia jovem no mercado mundial e ainda mais jovem no mercado nacional. Espera-se que com ela seja possível dar um salto considerável no comércio e no relacionamento eletrônico utilizando a Internet, o que parece ser uma tendência sem retorno tanto para empresas quanto para governos. Para que isso possa ocorrer é necessário que a tecnologia em questão, a de infra-estrutura de chaves públicas, evolua e se torne mais difundida e segura e com isso surjam novas aplicações e ocorra uma redução nos custos associados.

Essas questões são tratadas direta ou indiretamente nesta dissertação e espera-se que com os resultados obtidos seja possível:

• compreender as características principais dos produtos comerciais para implementação de ICPs e algumas arquiteturas onde estes produtos podem ser utilizados,

• verificar a capacidade de serem usadas algumas soluções *opensource* em implementações de ICPs com funcionalidade semelhantes às existentes nos produtos comerciais, com o diferencial da capacidade de adequação, expansão, correção de falhas e custo fornecida pelo tipo de licenciamento do código,

• comparar as funcionalidades de dois dos mecamismo de verificação de situação de certificados mais utilizados na atualidade e desta forma empregar cada um deles segundo suas melhores características, maximizando a segurança, a funcionalidade e a performance do sistema resultante,

• avaliar o impacto no tráfego da rede do transporte das LCRs e das trocas de mensagens associadas ao protocolo OCSP nos principais cenários de certificação

existentes atualmente,

• avaliar o esforço computacional necessário à geração, processamento e utilização dos certificados digitais, LCRs e mensagens do protocolo OCSP, fator de extrema importância tendo em vista a recente entrada em utilização de dispositivos de baixa capacidade de processamento, como celulares e computadores de mão em transações comerciais e bancárias,

• verificar a adequação e os benefícios da tecnologia de chaves públicas e certificados digitais para o controle de acesso a redes infra-estruturadas através de redes sem fio.

Esses resultados contribuem para uma melhor compreensão da tecnologia de infra-estrutura de chaves públicas e por conseguinte, uma maior facilidade na utilização da mesma não apenas no contexto apresentado, mas devido a documentação apresentada, também em outros contextos.

Além disso, a documentação dos procedimentos utilizados nas simulações, nos desenvolvimentos e nos próprios programas poderá ser utilizada como base para novos desenvolvimentos e pesquisas uma vez que a solução, como um todo, terá as seguintes características:

• código fonte em C/C++ portável;

• suporte aos padrões vigentes e ao estado da arte em checagem de certificados pela implementação de um cliente e um servidor OCSP;

• implementação multiplataforma (Windows e Linux);

• interface simples e amigável.

Os programas *opensource* analisados e os desenvolvidos poderão ser usados em projetos piloto de empresas que queiram avaliar a tecnologia assim como no meio acadêmico, como ponto de partida para pesquisas, uma carência atual visto que não foi encontrado em nenhum ambiente acadêmico ou mesmo comercial de testes, uma ICP com essas características.

Concluindo, esta dissertação irá contribuir no desenvolvimento e uso das ICPs visto que:

• Existe um longo caminho de padronização que ainda precisa ser trilhado. Muitas fases e processos empregados numa ICP ainda não estão padronizados e dificultam a interoperabilidade das soluções.

• O uso de projetos piloto pode fornecer o ferramental ideal para que testes e avaliações sejam desenvolvidos, por empresas e governos, visando estabelecer estes padrões ainda em aberto ou selecionar adequadamente um, quando for o caso.

• Ainda existe uma grande dificuldade de implementar aplicativos que usem certificação digital.

1.4 Trabalhos Correlatos

Durante os últimos anos o tema segurança em redes de dados tem se tornado cada vez mais importante, motivando trabalhos em diversas áreas relacionadas como na proteção de redes TCP/IP[14].

Os particularmente relacionados com a certificação digital, decorrentes de um período inicial marcado por visões gerais[15] podem ser classificados em algumas áreas maiores que caracterizam o estágio atual de desenvolvimento e são:

• implementação de protocolos[16][17] e algoritmos criptográficos[18][19][20],

• aplicações, como na votação eletrônica[21], na assinatura digital de documentos eletrônicos[22], na protocolização de documentos eletrônicos[23] ou na datação de documentos eletrônicos[24].

Já entre os relacionados com as redes sem fio, em especial sobre sua segurança, destacam-se[25] e [26].

1.5 Organização

Esta dissertação consiste de 3 partes, subdividadas da seguinte forma:

• Primeira Parte: Introdução Geral

Apresentada no capítulo 1 (Introdução) e no capítulo 2 (Infra-estruturas de Chaves Públicas)

• Segunda Parte: Desenvolvimento, Implementação e Testes

Disposta no capítulo 3 (Avaliação de Soluções para ICPs), capítulo 4 (Proposta de Implementação do Protocolo OCSP), capítulo 6 (Ambiente de Teste: AirStrike) e no capítulo 6 (Medições e Resultados)

• Terceira Parte: Considerações Finais

Disposta no capítulo 7 (Conclusões e Trabalhos Futuros)

Capítulo

2

Infra-estruturas de Chaves Públicas

O objetivo deste capítulo é fornecer os subsídios necessários para uma perfeita compreensão dos protocolos e procedimentos envolvidos na gerência e utilização dos certificados de chave pública, indo da requisição à revogação dos mesmos, passando pelas estruturas e padrões utilizados.

2.1 Introdução

O surgimento das ICPs não está bem claro na literatura especializada, mas o início da década de 90 parece ser a época mais provável. Em 1988 a ITU-T divulgou a primeira versão da recomendação X.500 e associada a ela a primeira versão do formato de certificado presente na recomendação X.509. Em seguida, em 1991, a RSA Security divulgou a primeira versão da família de padrões para criptografia de chave pública chamada PKCS. Aproximadamente na mesma época um algoritmos de chave pública, o RSA, foi incluído em um programa comercial, o Lotus Notes.

No início de 1993, a especificação do PEM (*Privacy Enhanced Mail* [35]) é divulgada e no mesmo ano a ITU-T apresenta, entre outras, a primeira revisão das recomendações da série X.50x e a RSA Security sua revisão dos PKCS com suporte ao PEM.

Pela cronologia e analisando o conteúdo dos documentos, é possível notar que tanto a RSA Security quando a ITU-T sofreram influências das idéias contidas na especificação do PEM, por isso, pode-se considerar o PEM como a primeira especificação de uma ICP funcional, padronizada e livremente disponível e desde então as ICPs passaram a ser foco de estudos e diversos novos padrões e produtos têm surgido.

Em uma ICP o objeto central é o certificado digital (detalhes de sua história e estrutura estão presentes no Apêndice I). Em torno dele gira todo um complexo sistema composto em parte por artefatos de software e em parte por procedimentos operacionais.

O funcionamento real de uma ICP depende intimamente de algumas decisões tomadas durante as fases de projeto da arquitetura do sistema. O número e as funções das entidades participantes, assim como a perfeita definição do processo de operação, são conseqüências dessas decisões.

A Figura 2-1 apresenta uma arquitetura genérica para uma ICP. Alguns componentes podem ser desdobrados em uma ou mais entidades funcionais por razões específicas de projeto baseadas nas funcionalidades específicas necessárias.

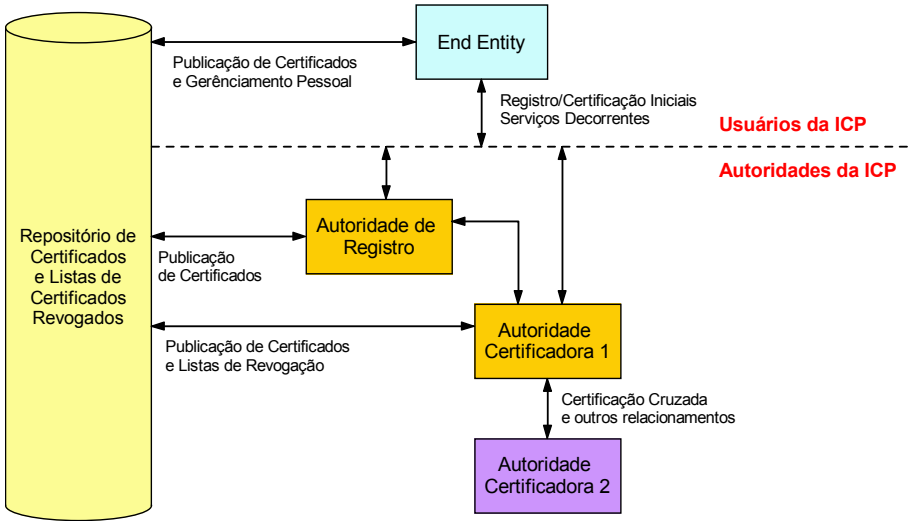


Figura 2-1. Arquitetura simplificada de uma ICP

Uma ICP consiste de três classes de componentes. São eles:

• as autoridades: as principais são as de registro e certificação. Em algumas implementações outras autoridade com funções auxiliares podem ser utilizadas;

• os clientes: separados em duas outras classes não mutuamente exclusivas: os detentores dos certificados, que os utilizam nas assinaturas de documentos, por exemplo, e os verificadores de certificados, que não possuem obrigatoriamente um certificado, mas querem validar uma assinatura, por exemplo. Na Figura 2-1 são

indiscriminadamente representados como *End Entity*;

• o repositório de armazenamento e disponibilização dos certificados, LCRs e informações afins.

As relações desses três tipos de componentes, desdobrados nos participantes básicos apresentados, descrevem o modo de operação de uma ICP. Interligando estas relações é que estão os padrões e protocolos necessários a tornar a comunicação a mais padronizada possível de modo que produtos de empresas distintas possam se integrar.

Algumas interações nesse sistema são bem comuns, como por exemplo:

• entre o detentor do certificado e o repositório: essa interação é necessária para possibilitar ao detentor o gerenciamento das informações definidas no âmbito da entidade de registro, ou seja, dados como endereço para correspondência, email para contato e outros que não fazem parte e não precisam ser alterados no certificado.

• entre o detentor do certificado e as autoridades: nesta interação o solicitante interage com as autoridades em diversos processos como na requisição dos certificados, no acompanhamento do processo de certificação e na obtenção do certificado do sistema.

• entre as autoridades e o repositório: em geral, todas as ações durante o processamento dos certificados pelas autoridades (incluindo a geração e divulgação das LCRs) precisam ser registradas e em muitos casos, também as ações dos clientes.

Uma descrição mais detalhada dos componentes apresentados na Figura 2-1 fornece mais detalhes sobre as responsabilidades e condições de operação, o que será feito na seção seguinte.

2.2 Componentes Principais

Dada a complexidade e a natureza distribuída do processo de geração, processamento e emissão de certificados, torna-se imprescindível a especialização de tarefas

fazendo com que uma ICP completamente funcional possa ser vista como pequenos sistemas interagindo entre si.

Para desempenhar seu papel, a infra-estrutura de chave pública deve ser capaz de realizar todo o processo de emissão de certificados, armazenamento, publicação (ou acesso on-line), revogação e arquivamento para verificação futura. Em conseqüência disso, esse sistema constitui um artefato computacional complexo, com capacidade de comunicação, processamento e armazenamento com requisitos muito específicos. Além disso, tanto as comunicações internas (entre componentes) como as externas (entre ICPs) desse sistema também devem ser seguras.

Dos participantes relacionados anteriormente, os três principais que precisam ser descritos e melhor compreendidos são:

- **Autoridade Registradora (AR[4]):**

A Autoridade Registradora atua recebendo as requisições de certificado e executando uma fase preliminar de checagem dos dados. Sua função principal é aliviar a carga da Autoridade Certificadora (AC) e tornar a natureza distribuida do processo viável ou seja, atua como a porta de entrada do sistema repassando para as outras partes as informações recebidas e disparando vários processos, quando necessário.

Sua presença é opcional em ambientes pequenos, mas quando adotada, geralmente possui dois pontos de acesso: o primeiro faz o papel de coletor de informações, implementado em muitos casos como um sistema web consistindo de um formulário e alguns scripts de apoio; o segundo é utilizado pelo pessoal da gerência da AR. Essa divisão torna-se importante para separar o operador da AR, que executa ações de gerência, em processos off-line, da parte de captação de informações, uma tarefa intrinsicamente on-line.

É importante destacar que os níveis de segurança da AC e da AR são bem diferentes (e conseqüentemente, os custos associados) o que torna mais viável e comum ter uma série de ARs executando um pre-processamento e depois fazendo o repasse para um pequeno conjunto de ACs.

4. Esta pode assumir diversos nomes na literatura internacional, como *Registration Authority* - RA, *Local RA* ou *Organizational RA* - ORA)

• **Autoridade Certificadora (AC):**

Estabelecer uma autoridade certificadora é uma responsabilidade que acarreta a administração de uma base de certificados, o estabelecimento de procedimentos técnicos e a criação de uma estrutura para o gerenciamento das chaves. Autoridades certificadoras não apenas emitem certificados mas também devem gerenciá-los, determinando a validade de cada certificado e as condições de renovação e possivelmente gerando e armazenando uma lista de certificados que já foram emitidos mas não estão mais válidos, ou seja, foram revogados.

Observando essa lista de atribuições, pode-se concluir que a responsabilidade da AC não termina após a emissão do certificado e nem após a sua revogação.

A emissão de um certificado deve, ao menos em parte, ocorrer de maneira off-line e não ser desenvolvida através de um mecanismo automático de solicitação/resposta. Antes da AC assinar um certificado, ela deve verificar os dados da solicitação, pois quando o certificado é gerado, a AC está garantindo que os dados do certificado são confiáveis.

É importante que a transferência das informações necessárias à emissão do certificado para a autoridade certificadora não seja comprometida e que a segurança física da AC seja garantida (se a chave privada da AC tornar-se pública, todos os certificados assinado tornam-se inseguros).

Finalmente, para completar a gestão de certificados, uma operação de revogação deve ser provida pela AC. Os certificados devem ser revogados pela AC e comunicados de alguma forma aos outros, sempre que:

• A chave secreta do usuário for comprometida.

• Os dados do usuário forem modificados. Por exemplo, o usuário mudou de

organização.

- O usuário não deseja mais ser certificado pela AC.
- O certificado da própria AC foi comprometido.
- O usuário violou a política de segurança da AC.

Deve-se observar que, depois de revogados, os certificados não deixam simplesmente de existir, passam somente a não ser mais válidos, podendo ser arquivados para efeito de comprovação futura.

- **Repositório de Certificados:**

Um certificado pode ter sido emitido pela AC, sanando a questão da associação entre chave e entidade, mas a menos que se possa localizar este certificado facilmente, ele efetivamente não seria diferente de um certificado que nunca tivesse sido criado e a ICP seria inútil.

Essa é a função principal do repositório: armazenar e tornar disponível os certificados e as listas de certificados revogados aos usuários da ICP. Para isso, algum tipo de repositório robusto, escalável e on-line deve existir.

2.3 Funcionamento Básico

Uma ICP básica possui ao menos 3 processos, sinteticamente representados na Figura 2-2, que apresentam o ciclo de vida de um certificado digital.

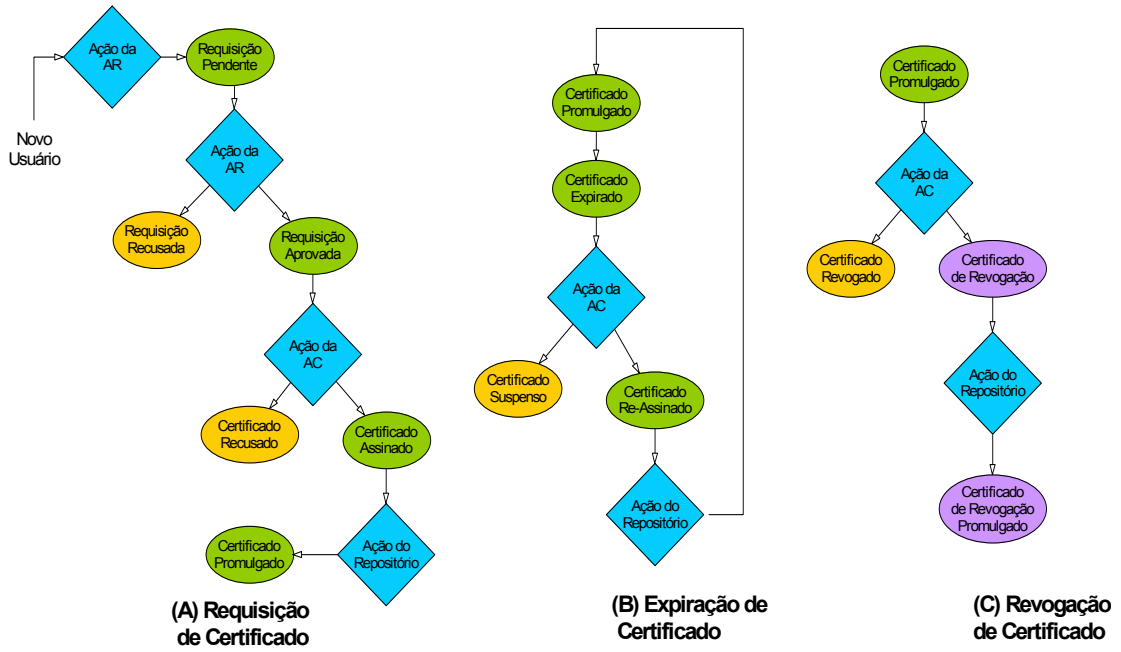


Figura 2-2. Síntese do ciclo de vida de um certificado digital

Em maiores detalhes os processos básicos são:

• Requisição do certificado

Representado na parte A da Figura 2-2, esse proceso inicia com uma ação do usuário ao solicitar o certificado. Normalmente ele o faz via formulário web processado pela AR. O resultado dessa ação é uma requisição de certificado num formato padronizado.

Durante esse processo é gerado, local ou remotamente, o par de chaves. A chave pública é incluída no formato da requisição do certificado. O formato também provê uma informação que pode ser usada pelo sistema e pelo usuário para acompanhar o processo de requisição.

Em seguida a gerência da AR verifica os dados da requisição e delibera sobre eles. Estando tudo de acordo, o formato é passado para a AC para que esta execute a última linha de verificações e assine o certificado. Finalizando o processo o certificado torna-se disponível, o que é feito publicando-o num repositório.

• Expiração do certificado

Representado na parte B da Figura 2-2, esse proceso não possui uma ação inicial definida. Ele começa quando o período de validade do certificado termina. Neste

momento o certificado passa ao estado de expirado e pode mudar de estado com uma ação da AC quando esta re-assina o certificado. Isso não ocorrendo, o certificado fica suspenso até que a AC emita um novo. Note que neste caso a segurança da associação não foi perdida, o que ocorreu foi o término de um contrato no qual a AC garantia o certificado, por isso a re-assinatura torna-se possível.

• Revogação do certificado

A revogação do certificado difere muito da expiração. A menos que os certificados tenham uma vida tão curta que sejam efetivamente usados uma única vez, quer dizer, eles são emitidos, são usados imediatamente, e nunca mais são usados novamente, alguma forma de revogação é requerida para situações nas quais um certificado deva ser declarado inválido pela AC.

A revogação ocorre quando uma entidade detentora de um certificado ainda dentro do período de validade desiste de possuir o certificado ou quando a AC detecta de alguma forma uma violação que torne a sua garantia de associação duvidosa ou inválida. Isso pode ocorrer quando a AC é informada ou detecta que a chave privada associada ao certificado em questão foi comprometida, ou seja, foi perdida ou furtada. Este é um processo importante e será descrito em detalhes na seção 2.5.

2.4 Modelos de Confiança

Em geral as ICPs são implementadas de forma a serem um espelho de estruturas organizacionais, refletindo uma hierarquia ou alguma forma de relacionamento onde direitos e deveres são compatilhados entre os participantes. Esses relacionamentos podem variar da simplicidade de uma relação de confiança direta (quando uma AC isolada assina os certificados de seus usuários) até níveis tão complexos quanto a estrutura de uma grande empresa.

O relacionamento entre usuários de uma mesma AC ou de ACs diferentes assim como entre ACs gera uma série de caminhos que foram caracterizados e chamados de modelos de confiança.

Um conceito importante é que o que chamar de confiança. No caso de certificados assinados por uma AC, podemos dizer que houve uma confiança direta, baseada nas

características operacionais, legais e outras. Quando uma AC assina um certificado de uma outra podemos pensar na relação do usuário com esta AC indireta como tendo uma confiança induzida.

Existem algumas topologias básicas[11] para os caminhos de validação. Nelas o sentido de validação é importante.

As fundamentais são:

• Estritamente Hierárquica: gráficamente representada por um árvore (vide Figura 2-3) tem a característica fundamental de propagar o sentido de confiança "de cima para baixo" e criar um caminho único de cada folha até o nível mais alto da árvore. Essa característica é muito desejável por ocasião da determinação do caminho de verificação, reduzindo o esforço de pesquisa e gerando sempre uma situação de certificados diferentes de indeterminado. A altura da árvore pode ser determinada, via campo de extensão nos certificados, tornando possível até limitar o tempo de procura. Essa estrutura rígida é bem adequada quando torna-se imprescindível um controle central e uma imposição de regras. Esta topologia foi a primeira a ser empregada, no padrão PEM, e pela sua rigidez causou o fracasso desse padrão. A idéia inicial de que existiria uma raiz única e mundialmente aceita foi utópica e hoje é tida como inviável.

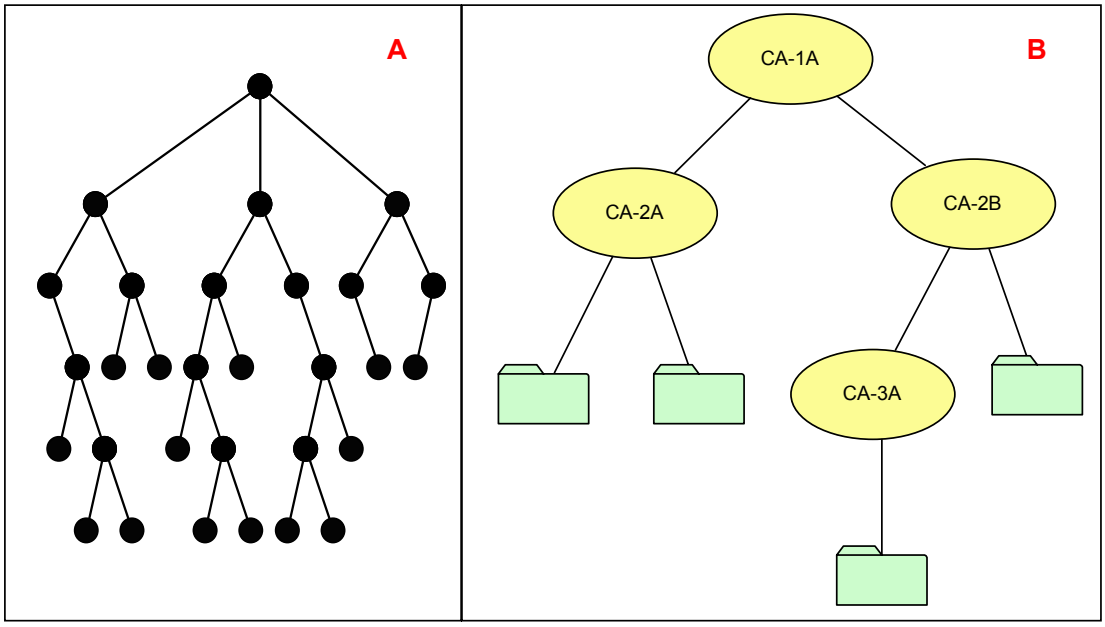


Figura 2-3. Topologia estritamente hierárquica

• Centrada no usuário: Nesta topologia não existe uma figura central coordenando alguma política ou ditando a hierarquia de nomes. É a base do funcionamento do PGP[27]. No PGP cada usuário, ou nó, decide em quem quer confiar e também quanto quer confiar. Para julgar a validade de um certificado o verificador aplica crivos próprios ponderando sobre quando ele confia e em quem, o que pode culminar em um resultado indeterminado. Esta é a única topologia que pode gerar esse resultado.

Gráficamente ela pode ser representada pelo exemplo da Figura 2-4.

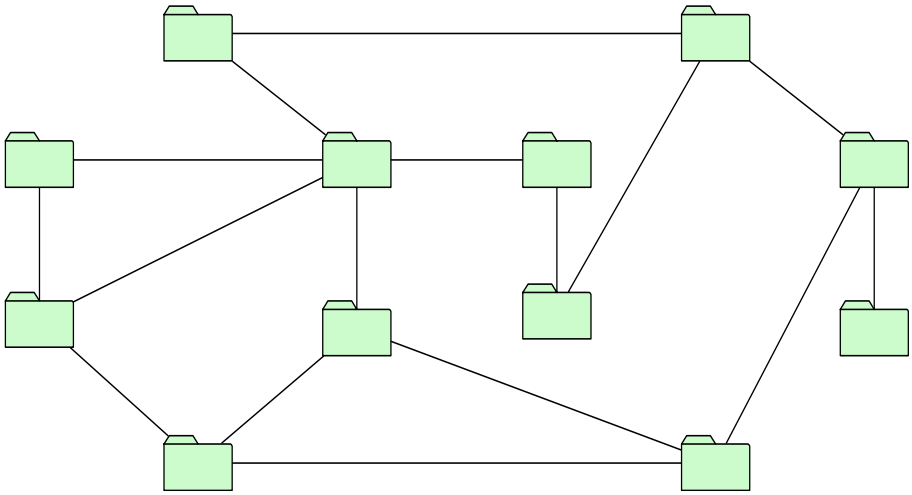


Figura 2-4. Topologia centrada no usuário

• Cruzada, em Ponte ou Malha: Estas topologias possuem como característica principal a interligação de domínios, hierárquicos ou não, de formas diversas. Com isso se consegue intermediar as características dos domínios conectados.
Estas topologias são freqüentemente adotadas por organizações com alto grau de complexidade.

Estas topologias são importantes na verificação da situação do certificado, como será visto em detalhes na próxima seção, pois após a obtenção do certificado digital desejado em um repositório seguem-se duas fases que culminam no fornecimento da situação do certificado, e que não depende apenas do certificado obtido, mas sim da cadeia à qual ele pertence.

Essas duas fases são: a determinação do caminho de validação e a verificação da validade ao longo deste caminho. Esses dois passos, ainda que sejam executados por algum servidor do sistema de gerenciamento ou pela estação do cliente, estão intimamente atrelados aos certificados pois são estes que carregam consigo as informações necessárias para a construção do caminho de validação assim como outras necessárias ao próprio processo de validação.

2.5 Revogação de Certificados

Como mencionado anteriormente, os certificados são usados para associar uma identidade ou alguma outra informação a uma chave pública. Geralmente essa associação perdura por toda a validade do certificado, mas pode ocorrer, por razões diversas, de neste período ser necessário cancelar esta associação.

Nestes casos torna-se fundamental a existência um método eficiente que permita ao resto da comunidade de possíveis usuários dos certificados localizar as informações necessárias sobre a situação de um certificado, possibilitando que o mesmo não seja usado após sua revogação. Estes procedimentos de quebra de associação é chamado de revogação de certificado e será abordado nesta seção onde os métodos principais serão analisados com relação à sua eficácia e aplicabilidade em alguns ambientes.

Retornando a questão dos certificados, estes uma vez emitidos nunca mais deixam de existir, ou seja, não se pode deliberadamente remover um certificado de um repositório devido a alguma falha na criação do mesmo ou por ocasião da sua revogação. Isto ocorre porque o certificado em questão pode já estar sendo usado e a ausência do mesmo no repositório irá criar uma inconsistência, podendo gerar uma desconfiança sobre o processo de emissão, invalidando a autoridade do emissor como realizador do processo.

O processo de emissão de um certificado ocorre, geralmente, independente do processo de revogação, devido principalmente a diferença semântica dos processos. Em comum, os dois possuem apenas o próprio certificado e a necessidade da divulgação das suas ações.

Essa diferença semântica entre os dois processos é de vital importância para a compreensão do processo de revogação.

Relembrando o processo de emissão, um certificado é primeiramente solicitado a uma autoridade, e após esta autoridadee ter se convencido das informações apresentadas, ela atesta este fato emitindo o certificado.

Deste ponto em diante outros agentes que desejem usar este certificado executam um processo de localização do mesmo, fazem cópia dele e passam a usa-lo até que a data limite da associação, estipulada pela autoridade, chegue.

Durante este período, pode ocorrer algum fato que motive a quebra desta associação. A questão que surge em seguida é: como informar a um número desconhecido e possivelmente grande de agentes que um dado certificado não pode mais ser considerado válido?

Não existe até o momento nenhum processo que, por ação da autoridade, informe a todos os detentores de certificados de uma comunidade que um ou mais de seus membros deixou de ser confiável. O procedimento adotado atualmente pela autoridade é disponibilizar de alguma forma, um processo pelo qual todo aquele que desejar checar se um certificado é valido possa fazê-lo.

A geração e divulgação da informação de revogação de um certificado, que são processos independentes, pode ocorrer de várias formas, como será visto, algumas delas muito semelhantes ao procedimento utilizado para divulgar um certificado emitido, porém estas formas não suprem todas as necessidades deste procedimento.

Por exemplo, no caso da localização de um certificado emitido, se o agente que procura o certificado não encontrá-lo a conseqüência disso é que ele não poderá executar alguma ação. Já no caso da não localização da informação sobre a revogação de um certificado, o agente ficará num impasse: usar o certificado correndo o risco do mesmo estar revogado ou não usar o certificado, mesmo que este possa estar válido.

Essa questão e outras relativas serão descritas e avalidadas sobre o ponto de vista do impacto causado, nas sessões que se seguem. Para tal, os mecanismos utilizados foram classificados em dois grupos maiores, e descritos a seguir.

2.5.1 Mecanismos Disponíveis

Como já mencionado, a verificação da situação do certificado possui duas fases distintas:

- a de localização do provedor da informação
- e neste servidor, a localização da informação desejada.

Em alguns mecanismos, o agente ainda precisa executar um processo para filtrar a informação obtida do provedor.

Os métodos existentes atualmente podem ser classificados em dois grupos maiores, que distingüem-se pela forma como a autoridade é contactada e como ela provê a informação sobre a situação dos certificados. Os métodos são os que se seguem.

2.5.1.1 Método de Publicação Periódica

Métodos de publicação periódica são técnicas caracterizadas pela divulgação de informações de revogação em períodos pré-determinados na forma de uma estrutura de dados assinada.

A relação que se segue é composta pelos mais utilizados atualmente, são eles:

• Lista de Certificados Revogados:

As LCRs (ou *Certificate Revocation Lists* - CRLs) são certificados que contêm uma estrutura de dados com uma lista seqüêncial de certificados revogados (apenas uma referência aos certificados na verdade, como um identificador único) e são sem dúvida o formato atualmente mais utilizado (detalhes sobre sua estrutura e evolução são apresentados no Apêndice I).

O local e o protocolo de acesso à lista é definido em um campo de extensão em cada certificado emitido. Este fato gera alguns problemas que serão apresentados a seguir.

A autoridade emissora da lista é geralmente a mesma que emite os certificados, entretanto, isto não é uma exigência crucial do processo, outra autoridade com relacionamento de confiança adequado com a emissora dos certificados pode emitir LCRs, isto pode ser visto em detalhes no mecanismo conhecido como LCR Indiretas[36].

Uma característica importante das LCRs é que devido ao fato de serem certificados digitais, muito semelhantes até na estrutura dos certificados de associação (vide Apêndice I), podem ser armazenadas (localmente ou em servidores de *cache*) e transportados sem a necessidade de segurança extra no processo. A assinatura da estrutura garante a integridade dos dados. Entretanto, como todo certificado, ela precisa ser checada, pois a autoridade que assinou a lista pode não ser válida.

Atualmente a maioria dos certificados deste formato obedecem a versão 1 da estrutura. Essa versão possui problemas de escalabilidade, funcionalidade e segurança que foram corrigidos pela versão seguinte. Infelizmente ainda é possível encontrar muitos sis-

temas usando a primeira versão devido principalmente à compatibilidade com os clientes existentes.

Na versão atual (a segunda) foram introduzidos campos de extensões, que podem ser marcados como críticos ou não, e que existem em dois contextos diferentes, que são:

- extensões para a lista, com informações válidas para toda a lista,

- extensões por certificado revogado presente na lista, com informações extra sobre cada certificado revogado presente na lista.

As extensões marcadas como críticas precisam ser compreendidas e processadas pelo cliente, já as não-críticas podem ser simplesmente ignoradas pelos clientes que não compreenderem sua semântica.

Devido à estrutura de dados usada nas LCRs, algumas variações foram geradas principalmente para evitar o sério problema de escalabilidade. Esse problema é devido à dificuldade que recairia sobre os clientes caso a estrutura fosse particionada e estes tivessem que recompô-la. Esse particionamento surge como uma solução quando as LCRs tornam-se grandes, o que pode ocorrer devido às políticas de certificação da autoridade ou devido à falha no controle do tamanho da lista. É importante ressaltar o fato da LCR nascer com o primeiro certificado emitido e perdurar por toda a existência da autoridade. Ou seja, sem os devidos controles, essa estrutura não deixará de crescer. Com o aumento do número de certificados emitidos e conseqüentemente do número de certificados revogados, em algum momento o tamanho da lista, assim com o seu transporte, passarão a causar problemas de desempenho da rede e até mesmo do cliente, uma vez que este, após obter a lista, ainda terá que pesquisar nela por sua própria conta para verificar se o certificado que deseja verificar está ou não na lista.

Um outro problema associado com o anterior relaciona-se com a freqüência de geração da lista. Toda a lista gerada informa (em um campo obrigatório da lista) quando a próxima estará disponível, ou seja, o momento geração da próxima é determínistico e independente de ocorrerem centenas de revogações ou nenhuma nesse período.

Analisando os dois extremos desta situação vemos que:

1. pode-se fazer o intervado entre as listas tender a zero e neste caso o cliente terá a melhor informação sobre a situação dos certificados, porém a um custo elevado devido à freqüência de *download* da lista.

2. de forma oposta, se o intervalo tender a infinito, o cliente não precisará obter as listas tão freqüentemente, ficando liberado desta tarefa, porém terá a pior informação sobre o estado dos certificados.

Devido à aleatoriedade das revogações, a prática comum é definir um intervalo fixo para as emissões que é tido como aceitável para uma classe de certificados.

Pode-se perceber que em ambos os casos ocorre um disparo de sincronismo, ou seja, após a data de expiração da lista todos os cliente irão solicitar uma lista nova a um mesmo servidor, gerando uma sobrecarga de acesso.

Algumas propostas atuais visam resolver um ou mais problemas das LCRs completas. É importante ressaltar que a grande maioria dos clientes existentes na atualidade ainda não suporta estes avanços, como é o caso dos produtos da Microsoft: o Internet Explorer e o Outlook.

As propostas presentes em [36] são:

1. alterar o par caminho e formato da lista: o formato atual, que especifica de uma forma única o protocolo e o caminho direto até a lista completa, passará a informar o caminho junto com outras informações usadas para localizar uma LCR não mais completa, mais fragmentada em partes menores, que facilitariam o gerenciamento (método conhecido como *CRL Distribution Point*). Uma sugestão para a fragmentação é o uso de LCRs incrementais (Delta CRLs). Uma lista base com todos os certificados é emitida com certa freqüência, e durante este intervalo apenas informações incrementais a esta lista são geradas.

2. que o caminho definido no certificado não seja direto ao local das listas, mas sim um apontador que pode ser usado para direcionar o cliente para um sistema mais adequado, baseado em algum parâmetro e que o particionamento das listas seja feito não mais usando tamanhos fixos.

3. que sejam usadas *Redirect CRLs* que fornecem uma método pelo qual pode-se conectar às LCRs dado um caminho até a primeira.

4. A única proposta que altera a estrutura de dados da LCR foi feita pela empresa Valicert e é conhecida como *Certificate Revocation Trees* (CRTs)[37]. As CRTs são baseadas nas árvores de hash de Merkle[38] onde a árvore em si representa toda a informação relevante sobre revogação de certificados de uma comunidade de autoridades certificadoras.

2.5.1.2 Métodos de Descobrimento On-line

Os métodos chamados on-line diferem dos de publicação periódica apresentados anteriormente em diversos aspectos, principalmente pelo fato de exigirem que as partes estejam on-line durante todo o processo.

O mecanismos mais popular desta classe e até o momento o único com alguma utilização é o OCSP (*Online Certificate Status Protocol*) definido na RFC 2560. Este protocolo implementa um método relativamente simples de requisição/resposta oferendo um caminho para obter em tempo real informações sobre a revogação de certificados de uma autoriade confiável referenciada com OCSP Responder.

A geração da requisição deste protocolo pelo cliente pode receber como entrada até quatro certificados, dois obrigatórios e dois opcionais, que são:

- o certificado a ser verificado,
- o certificado da autoridade que emitiu o certificado a ser verificado
- o certificado do OCSP *Responder*, que pode ser omitido caso seja o mesmo da autoridade certificadora. A presença de um dos dois certificados é obrigatória para validar a resposta, evitando que um *Responder* falso atue.
- o certificado do cliente que executa a requisição, que é usado quando a requisição precisa ser assinada, uma vez que certos *Responders* não respondem a pesquisas anônimas.

A Figura 2-5 resume o processo.

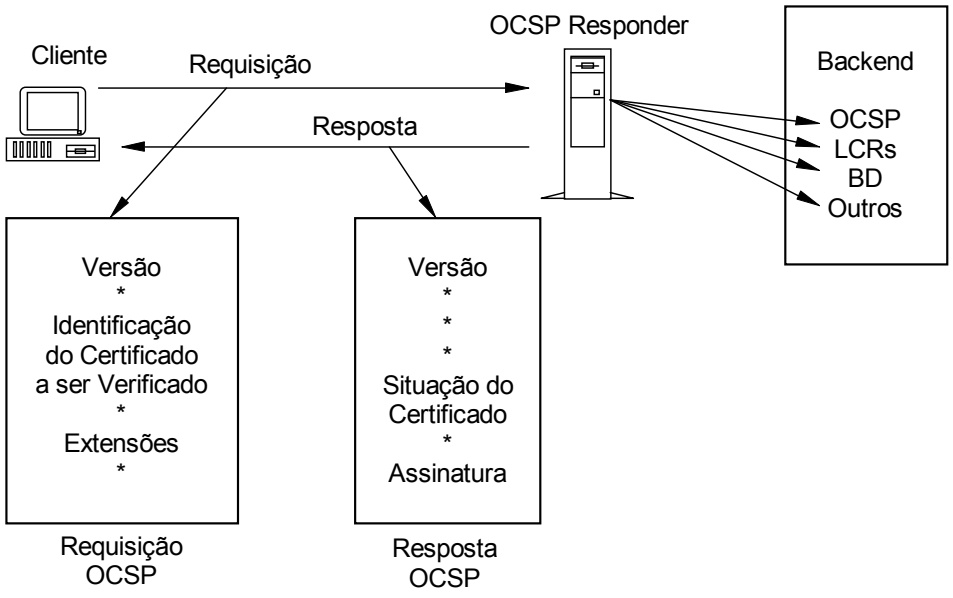


Figura 2-5. Interação cliente-servidor no protocolo OCSP

Outros protocolos[39] desta mesma classe foram propostos, porém ainda estão em estudo.

A tabela 2-1 resume as características de cada uma das soluções apresentadas.

Esquema	Síntese	Comentários
LCRs Completas	Estrutura de dados assinada contendo uma lista de certificados revogados	Largamente adotada, porém severamente criticada devido aos problemas de desempenho, escalabilidade e periodicidade, entretanto, alternativas baseadas nelas existem, mas ainda em estudo
Ponto de Distribuição de LCRs	Um método já padronizado porém ainda não adotado para o particionamento de LCRs	Solução com ganhos reais no desempenho e na escalabilidade porém a questão da periodicidade das listas ainda é problema. Também não possuem suporte nos clientes mais comuns, como os navegadores web.

Tabela 2-1. Sumários do mecanismos de verificação de certificados

Esquema	Síntese	Comentários
Delta CRLs	Um método já padronizado de divulgação de informações de revogação sem requerer uma CRL completa ou uma atualização do Ponto de Distribuição de CRLs	Pode ser usada em conjuto com os Pontos de Distribuição de LCRs para melhorar as questões do desempenho, escalabilidde e periodicidade das LCRs completas, mas sofrem do mesmo problema de aceitação do anterior.
LCR Indiretas	Outro método padronizado que permite informações sobre revogação de multiplas ACs residirem juntas numa única LCR	Emprego adequado quando o uso de várias LCRs de ACs diferentes gera problemas de perfornace. Ainda não são suportadas pelos clientes web.
OCSP	Uma forma on-line, baseada em mensagens simples, de obter a situação de um certificado	Ainda que seja capaz de fornecer respostas em tempo real é intimamente dependente da fonte geradora das informações de revogação, que podem não ser tão atuais quanto se deseja.
LCRS Redirecionadas e Referênciadas	Conceito relativamente recente, com suporte ao particionamento dinâmico de CRLs bem como multiplos métodos de recuperação de informações de revogação.	Conceito esperado para ser inserido na próxima versão do padrão PKIX
CRTs	Um tecnologia, definida pela Valicert, que provê uma forma de armazenar informações sobre revogação em baixo volume de dados através de uma estrutura baseada em árvore de hash binária.	Com potencial de vir a ser uma alternativa viável para a representação das informações sobre revogação baseada em serviço de terceiros. Ainda sem suporte nos clientes mais usados.

Tabela 2-1. Sumários do mecanismos de verificação de certificados

Capítulo 3

Avaliação de Soluções para ICPs

Tendo como foco um dos objetivos desta dissertação, foi feita uma análise de alguns produtos atualmente disponíveis para a implementação de uma de infra-estrutura de chaves públicas, tanto comerciais quanto *opensource*. Isto foi feito visando a caracterização das soluções comercialmente existentes e a escolha de um produto gratuito para ser usado na implementação de avaliação.

Na tarefa de analisar as opções existentes, existem algumas abordagens eficientes para caracterizar uma infra-estrutura de chaves públicas. Pode-se isolar apenas os aspectos técnicos e do mesmo modo somente os comerciais. A abordagem adotada nesta análise destaca principalmente os aspectos técnicos e funcionais dos produtos encontrados.

Considerando somente o sistema computacional necessário a uma infra-estrutura de chaves públicas e com base nas pesquisas realizadas é possível resumir a estrutura destes sistemas na Figura 3-1.

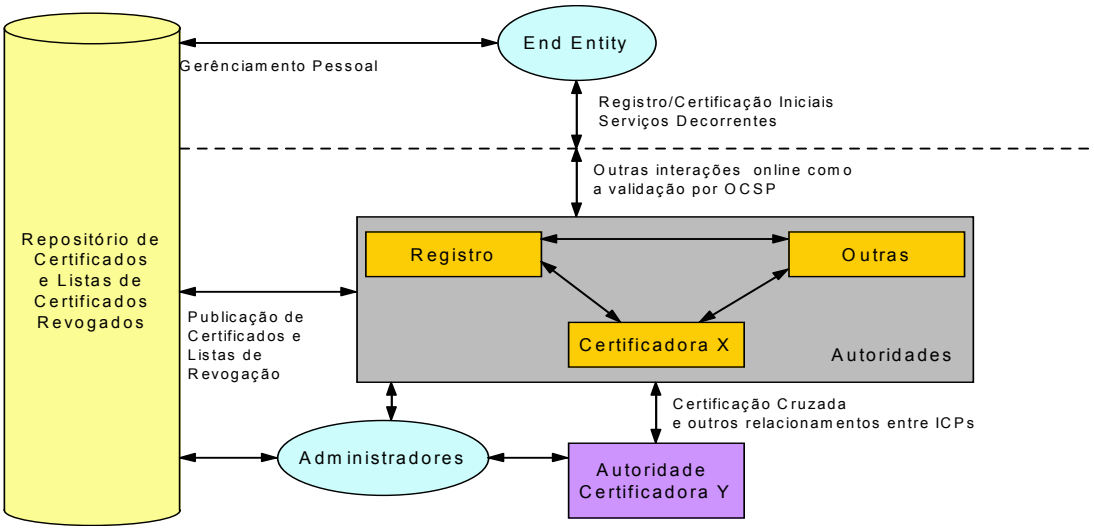


Figura 3-1. Componentes básicos de uma ICP

Cada uma das soluções analisadas realiza os componentes e as interações apresentadas na figura de forma a se adequar a um dado nicho de mercado, provendo algum tipo de funcionalidade que se torne um diferencial competitivo. Algumas dessas funcionalidades são descritas a seguir.

3.1 Soluções Comerciais

3.1.1 Empresas Analisadas

Devido à recente entrada no país desta tecnologia, ainda não existem muitas empresas comercializando tais soluções. Este fato também pode ser observado no mercado mundial.

O que se encontra, e com certa facilidade, são empresas comercializando soluções que fazem uso de certificados digitais e conseqüentemente dependem de uma ICP já estabelecida.

As empresas que comercializam soluções para ICP agem de duas formas: vendendo a solução em software ou prestando o serviço. A relação que segue apresenta cada uma das empresas e produtos encontrados no mercado nacional e a forma de operação.

- **UniCERT do Brasil Certificadora**[1]

Com sede em Porto Alegre, a empresa que se apresentou como a primeira certificadora que além de ser de capital nacional opera todo o processo no país (a Certisign antes de se filiar à ICP-Brasil mantinha o processo nos EUA), entrou em operação em 01/01/01 e é resultado da união da American BankNote Company (com 75% do capital e controladora da empresa de *smart cards* Gemplus) e da Telsul Telecomunicações (com 25 % do capital).

Seu principal *datacenter*, orçado em US$ 2 milhões (75% deste valor gasto na sala cofre) fica em local não divulgado de Porto Alegre e além desse a empresa possui dois outros como espelhos: um no Rio de Janeiro e outro nos Estados Unidos. O investimento inicial em toda a estrutura chega a US$ 5 milhões e também será aproveitado para outros dois serviços como armazenamento de bancos de dados e

1. Informações obtidas no próprio site da empresa, no endereço www.unicert.com.br

identificação biométrica.

Possui como parceira, com participação nos lucros e fornecedora da tecnologia, a Baltimore Américas, responsável por 60 % das operações de certificação dos Estados Unidos e com aproximadamente 85% do mercado no mundo. Seu sistema possui classificação de segurança E3 pelo ITSEC (*Information Technology Security Evaluation Criteria*)

O modelo de negócio da empresa, que esta focada no mercado nacional é uma espécie de ASP (*Application Service Provider*) para o setor de certificação digital. Até a presente dada não está filiada à ICP-Brasil e nem consta na lista das solicitantes de credenciamento.

- **Certisign Certificadora Digital**[2]

Os serviços e produtos da multinacional VeriSign são oferecidos no Brasil exclusivamente pela CertiSign Certificadora Digital S.A., única afiliada brasileira da VeriSign Trust Network, a sua rede mundial de confiança.

Possui um *datacenter* no Rio de Janeiro contruído em um local especialmente projetado e mantido para este fim e presta um serviço denominado ICP Gerenciada, semelhante ao modelo da UniCert, além de vender certificados com selo próprio para sites seguros e correio eletrônico em dois níveis de segurança. Possui o diferencial de ter o seu certificado raiz já disponível no Internet Explorer e estar credenciada junto à ICP-Brasil e por isso pode também emitir certificados compatíveis com este selo. No último semestre de 2003 foi homologada para emitir e-CPF e e-CNPJ que são certificados eletrônicos definidos no âmbito da ICP-Brasil para operações entre empresas e cidadãos e o governo .

- **RSA Security**[3]

Esta é de uma das três grandes empresas encontradas que comercializa a solução em software. A suite completa possui diversos módulos que interagem com o produto central (o RSA *Keon Certificate Authority* atualmente na versão 6.5), e além

2. Informações obtidas no site da Certisign, em www.certisign.com.br e na página da Verisign, em www.verisign.com

3. Informações obtidas no site da empresa, em www.rsasecurity.com e nos white papers disponíveis sobre o produto.

disso existe uma API completa para a criação de outros módulos.

O RSA Keon é composto por quatro módulos principais: o *Web Server*, o *Logging Server*, o *CMP Server* e o *Secure Directory Server*.

O *Web Server* é baseado no servidor web Apache e provê a interface primária do Keon CA.

O *Logging Server* armazena de forma segura (usando assinatura digital) os eventos do sistema fornecendo o material necessário para uma auditoria, que pode ser feita com o módulo *Keon CA Auditor*. Os logs são armazenado em arquivos locais e digitalmente assinados.

O *CMP Server* é empregado no processo de requisição dos certificados e nas tarefas relacionadas à administração do processo.

O *Secure Directory Server* é um servidor de diretórios LDAP que armazena todas as informações da ICP, como os certificados e as LCRs.

- **Baltimore**

A solução da Baltimore, chamada Unicert e atualmente na versão 5 e utilizada pela empresa Unicert do Brasil, possui uma arquitetura desenhada para ser a mais flexível possível como o objetivo de atender a uma ampla faixa de requisitos de negócio e legais.

Os módulos que compõem o sistema podem ser instalados em uma mesma máquina ou em equipamentos separados para evitar problemas de desempenho nas funções e neste caso toda a comunicação ocorre via conexões TCP/IP seguras.

Os módulos principais do sistema são:

- UniCERT *Certification Authority*: é o núcleo da solução e é responsável por gerar e assinar os certificados e as LCRs.

- UniCERT *Certificate Status Server*: este módulo provê informações em tempo real da situação dos certificados usando o protocolo OCSP para todos os módulos do UniCERT assim como para requisições externas.

- UniCERT *Publisher*: a AC tipicamente necessita tornar público os certificados emitidos. No UniCERT isto é feito por este módulo que possui a capacidade de exportar os certificados para vários formatos e publicá-los em sistemas LDAP de

diversos fabricantes além do Microsoft Active Directory. Como diferencial desse sistema, existe a capacidade de controlar quais certificados serão publicado em quais sistemas.

• **Entrust Authority Security Manager**[4]

A suite de produtos da Entrust tem como parte central o Entrust *Authority Security Manager* atualmente na versão 4.5. Este sistema é rodeado por uma série de outros, que são:

- *Self-administration Server*: responsável por automatizar e assim facilitar todo o processo de requisição de certificados e recuperação de chaves.

- *Administration Services*: aplicação baseada na web que possibilita a delegação e distribuição da administração do Security Manager

- *Timestamp Server*: utilizado para fornecer um selo temporal para as ações realizadas

- *Enrollment Server for Web*: empregado para emitir certificados digitais para aplicações e dispositivos que operem via web

- *Enrollment for VPN*: versão do sistema anterior voltada para aplicações e dispositivos que utilizem VPNs.

3.1.2 Soluções Analisadas

As empresas que fornecem soluções de ICP como serviço, não puderam ser avaliadas pois nenhuma informação sobre como seus sistemas operam ou são implementados é fornecida. Estas informações são guardadas como segredo de negócios e todo contato feito visando obter informações ou foi ignorado ou recebeu como resposta o texto padrão: "... essa informação não pode ser fornecida". A seção 3.1.2.2 apresenta uma análise comparativa das soluções de ICP fornecidas como serviços e das fornecidas como produtos.

As três empresas que vendem o produto (a RSA, a Entrust e a Baltimore) fornecem em suas páginas eletrônicas informações e artigos técnicos que apresentam um conjunto bem completo de informações sobre seus produtos. Isto inclui informações de como eles

4. Referenciada nas tabelas de características como Entrust ASM por simplicidade

podem ser implantados (incluindo como integrá-los com outras aplicações) e alguns *cases* reais. Nelas é possivel verificar que existe uma estreita relação entre os componentes genéricos da Figura A.1 com os existentes em cada produto. As funcionalidades básicas diferem pouco, variando, em geral, no nome atribuído, na forma como são implementadas (por exemplo, com relação aos ajustes das interfaces) e na forma de segmentação dos serviços.

3.1.2.1 Características

As informações apresentadas nas tabelas que seguem foram obtidas das páginas eletrônicas dos fabricantes, dos artigos técnicos por eles fornecidos sobre os produtos, dos *cases* implementados por algumas empresas que adquiriram a solução e em revistas eletrônicas especializadas neste nicho de mercado.

É importante destacar que essas informações foram encontradas pulverizadas por estas fontes, e em alguns casos, ocultas em detalhes das descrições.

As características foram agrupadas da seguinte forma:

Tabela 3.1: Sistema Operacional e Repositórios suportados[a]

Produto	Compaq Tru64	Microsoft Win NT4	Microsoft Win 2K	Solaris 7 e 8	HP UX 11.0	AIX 4.3.3+
Entrust ASM	Oracle	Informix	Informix	Informix ou Oracle	Informix	Informix
RSA Keon	N/A	LDAP Nativo	LDAP Nativo	LDAP Nativo	N/A	N/A
Baltimore UniCert	N/A	Oracle	Oracle	Oracle	Oracle	N/A

a. Nenhuma das plataforma fornece suporte a qualquer distribuição do GNU/Linux ou suporte a qualquer SGDB gratuito, como MySQL ou Postgress

Tabela 3.2: Métodos de Revogação de Certificados

Produto	CRL	OCSP	CRT[a]	CRL Distribution Point
Entrust ASM	Sim	apenas via produtos de terceiros	Via produtos de terceiros	Sim, e em conjunto com CRL única
RSA Keon	Sim	apenas via produtos de terceiros	Não	Sim

Tabela 3.2: Métodos de Revogação de Certificados

Produto	CRL	OCSP	CRT[a]	CRL Distribution Point
Baltimore UniCert	Sim	nativo e via produtos de terceiros	Via produtos de terceiros	Sim

a. CRT - Certificate Revocation Tree

Tabela 3.3: Segurança das Comunicações

Produto	Entrust ASM	RSA Keon	Baltimore UniCert
Comunicação com Clientes	via plug-in no cliente (Entrust/Entelligence), protegido por seções com SPKM/GSS-API e usando o protoclo PKIX-CMP	seções protegidas via SSL v2 ou v3	Vários, PKCS#10/7, PKCS#12, PKIX-CMP, SCEP
Comunicação AC-AR	seções SPKM/GSS-API e usando o protoclo PKIX-CMP	seções protegidas via SSL v2 ou v3	Mensagens PKIX CMP (sempre assinada)
Hard e Soft para guarda dos certificados?	Suportados como recurso adicional a segurança já existente	Sim. Usuários podem usar o RSA SecurID para obter seus certificados.	Sim para AC, AR e Admins, dependendo do controle de acesso estabelecido.
Chave da AC protegida por Hardware?	Suportados como recurso adicional a segurança já existente, dispositivos usados incluem: Chrysalis, Zaxus e Atalla	Sim, via HSM. Acesso controlado por segredo compartilhado.	Sim, via um dos seguinte modulos: Baltimore Technologies Sureware Keyper e nForce

Tabela 3.4: Característica da Autoridade Certificadora

Característica	Entrust Authority Security Manager	RSA Keon	Baltimore UniCert
Replicação do Repositório	Sim	Sim	Sim
Customização do Certificado por Cliente	Sim	Sim, dentro das extensões suportadas	Sim, completamente
Validações de Terceiros	CCE[a] EAL-3 FIPS 140-1 níveis 1 a 3	CCE EAL4+ e Identrus	CCE EAL4,FIPS 140-1 níveis 2, 3 e 4
Funções para Manutenção do Repositório (checagem de integridade, encriptação, etc)	Sim	N/A	Não

Tabela 3.4: Característica da Autoridade Certificadora

Característica	Entrust Authority Security Manager	RSA Keon	Baltimore UniCert
Operações sobre Volumes (relatórios, monitoramento e etc)	Sim	Sim	Sim
Extensões de certificados Suportadas (S/MIME, PKIX, IPSec, SSL)	Todas + PEM e SPKM	Todas + SET	Todas + PEM, SET e SPKM
API para integração com outros Sistemas	Java	C/C++ e Java	C++ e Java
Repositório encriptado	sim, via AES	Não	Não
Número Máximo de Usuários por AC	5 milhões (dados do fabricante)	8 milhões (dado de teste independente)	teoricamente ilimitado (informação do fabricante)
Número Máximo de ACs por Servidor	25(dados do fabricante)	indefinido (informação do fabricante)	ilimitado (informação do fabricante)

a. Common Criteria Evaluation

Tabela 3.5: Topologias Suportadas pela ICP

Produto	Entrust Authority Security Manager	RSA Keon	Baltimore UniCert
Certificação Cruzada Permitida	peer-to-peer e hierárquica. Redes hibridas também são suportadas. Suporte a PKCS 7/10 e PKIX-CMP	Via PKIX-CMP, PKCS#7 e PKCS#10.	Via PKCS#10 e PKCS #7
Em que nível a AC pode ter certificação cruzada?	Apenas raiz	Qualquer	Qualquer
Multiplas AC/ AR permitidas?	Sim, sem limites. As conexões com a AC são apenas limitas pelo S.O.	Sim, teoricamente ilimitados	Sim, em qq profundidade de hierarquia com ilimitadas ARs por ACs

Tabela 3.6: Suporte a SmartCard e/ou Token

Produto	Entrust Authority Security Manager	RSA Keon	Baltimore UniCert
Quais dispositivos e padrões são suportados ?	Para smartcards: PKCS #11 v1 e v2. Para biometria: BioAPI.	O RSA SecurID token é suportado via Virtual Smartcard. O sistema também suporta smartcards comuns nos padrões PC/SC e PKCS#11	Via PKCS#11, exemplo: Sureware Keyper, Chrysalis Luna CA3, nCipher nShield/ nForce, Datakey, Gemplus, Oberthur, ActivCard, Rainbow etc
Proteção do Cliente	Aplicações Entrust-Ready tem acesso ao Entrust Digital ID armazenado em smartcards e usam dispositivos biométricos para autenticação.	Sim, usando qualquer um dos citados acima	Específica os dispositivo que podem ser usados, mas normalmente emprega login e senha
Proteção do Administrador da AC	Uso de dispositivos de hardware citados anteriormente. Necessidade de acesso físico para certas operações.	Certificado de Administrador protegido pelos método acima.	Software / smartcard / token
Proteção do Administrador da AR	Uso de smartcards para autenticação da AR é suportado	Idêntica a anterior.	Software, smartcard ou token

Tabela 3.7: Gerenciamento de Chaves

Produto	Entrust Authority Security Manager	RSA Keon	Baltimore UniCert
Atualização Automática das Chaves ?	Sim. Tanto dos certificados com das chaves e sem a necessidade de intervenção do usuário ou do administrador de acordo com a política definida pela AR.	Não na versão corrente, o usuário precisa acessar uma homepage para atualizar suas chaves e certificados.	Não, planejado para ser suportado na próxima versão do sistema
Histórico Automático da Chaves ?	As chaves são gerenciadas automaticamente. O usuário não necessita saber qual chave foi usada num processo.	O RSA Keon Web PassPort permite que o usuário mantenha armazenado todo o histórico da chave em um smartcard virtual.	Depende do software do cliente

Tabela 3.7: Gerenciamento de Chaves

Produto	Entrust Authority Security Manager	RSA Keon	Baltimore UniCert
Backup e Recuperação da Chave ?	Backup apenas para chaves de sigilo, não para as de assinatura (identificação) A recuperação pode ser via auto-serviço baseado em segredo compartilhado sem a ação do admin da AR ou via ação dos admins da AR (requer multiplas aprovações para tal fim).	A AC pode ter um *Key Recovery Module* para as chaves de sigilo e faz uso de um HSM. Para AC com chaves baseadas em soft o backup é feito junto com o do sistema. Já as baseadas em hard necessitam de outro equipamento com tal funcionalidade.	Sim, através do *Key Archive Server*

Tabela 3.8: Interface de Gerenciamento

Produto	Entrust Authority Security Manager	RSA Keon	Baltimore UniCert
Interface de administração	Interface gráfica e por linha de comando.	Apenas Web	Apenas Gráfica
Multiplos níveis de Admins por AC ?	Sim, pode default existem 3 níveis.	Sim	Sim
Multiplos Admins por AR	Sim. Ilimitados.	Sim	Sim
Diferentes Admins podem ter diferentes tarefas associadas ?	Sim. Admins podem ser definidos com base em mais de 100 regras customizáveis, muitas destas já predefinidas.	Sim, suporte a regras administrativas definidas pelo Common Criteria.	Sim, operadores da AC podem ter separação de regras. Operadores das ARs so podem fazer uso de políticas a eles alocadas.

Tabela 3.9: Custo Aproximado por Licença e Recursos

Licenças / Recursos[a]	Entrust Authority Security Manager	RSA Keon	Baltimore UniCert
Suíte Básica para 100 licenças	39.350[b]	6.300[c]	64.800
Todos os extras para 100 licenças	15.000[d]	100[e]	3.500[f]
Suíte Básica para 1.000 licenças	81.750	34,170	79.200
Todos os extras para 1.000 licenças	34.500	1.000	3.500

Tabela 3.9: Custo Aproximado por Licença e Recursos

Licenças / Recursos[a]	Entrust Authority Security Manager	RSA Keon	Baltimore UniCert
Suíte Básica para 10.000 licenças	343.750	242.500	163.800
Todos os extras para 10.000 licenças	152.500	10.000	3.500

a. Preços em dolares americanos
b. Entrust/PKI + Entrust/Direct
c. Web Passport + Keon CA
d. Entrust/Roaming + Entrust/AutoRA
e. Keon RA apenas
f. Oracle + LDAP Server

3.1.2.2 Comparação entre as Arquiteturas Encontradas

Durante a pesquisa feita, duas arquiteturas de ICP foram detectadas: uma focada na prestação do serviço e outra na comercialização da solução em software. Cada uma dessas abordagens possui seus prós e contras, que podem ser resumidos nos seguintes:

• Prestação do Serviço: nessa opção a vantagem principal se encontra no baixo custo inicial de adoção da tecnologia, visto que não será necessário gerar (e em alguns casos até construir) um ambiente seguro nem treinar uma equipe para instalar, configurar e manter o sistema. Entretanto, isso pode causar um problema de dependência tecnológica e com a entrada em amplo uso da tecnologia pela empresa que contrata o serviço o custo pode vir a ser um problema de médio a longo prazo. Em geral o serviço é fornecido sobre a forma de venda por um período determinado de certificados, ou em casos onde o número de certificados e os controles sobre os mesmos precisam ser específicos pode-se optar por uma outra modalidade chamada de ICP Gerenciada onde toda a parte física e lógica permanecem sob o controle da prestadora do serviço e apenas uma interface de gerenciamento é disponibilizada ao cliente, onde este pode executar um maior número de configurações e ajustes. Este tipo de relacionamento pode ser inviável quando os controles necessários estipulados pelos clientes passam a ser muito rígidos e com valores de indenizações muito altos devido ao teor do negócio onde os certificados serão empregados. Nesta situação a opção mais adquada é a criação de um ambiente seguro próprio e a aquisição de uma solução em software.

• Venda da Solução: a aquisição de uma solução em software tem como dife-

rencial sobre a anterior o fato de garantir de certo modo uma independência tecnológica e uma autonomia sobre todo o processo. Em contra partida, somente a solução em software não é suficiente para implantar uma ICP, há a necessidade de um ambiente físico seguro e de uma equipe altamente qualificada para gerenciar todo o processo, o que eleva em muito o custo inicial dessa abordagem.

A implementação de um ambiente computacional de apoio a tecnologia da informação envolve diversas áreas do conhecimento que possuem requisitos e tratamentos distintos. A crescente necessidade desses sistemas conduziu ao desenvolvimento desta área de estudo e levou à criação de algumas normas para auxiliar na difícil tarefa de reunir equipamentos, pessoal e procedimentos de forma a obter um resultado desejado.

Uma ICP é em última instância um ambiente computacional que sofre com todos os problemas comuns a estes ambientes e por este motivo sua implantação pode se basear nas recomendações genéricas das normas existentes, como a NBR ISO/IEC 17799.

Existem duas soluções que podem ser adotadas para colocar em prática os controles físicos apresentados nas normas. A primeira é a construção de um local específico para abrigar os equipamentos e as operações. No país, apenas duas empresas constroem ambientes seguros de TI (incluindo prédios e salas-cofre) que são:

- Aceco TI (www.aceco.com.br) que utiliza a tecnologia empresa Alemã Lampertz
- Caviglia (www.caviglia.com.br) que utiliza a tecnologia da Americana Firelock

Esse domínio das duas empresas fica ainda mais claro quando observamos as normas mundiais existentes sobre ambientes físicos seguros. São apenas duas: uma Alemã chamada VDMA 2491 e outra americana, chamada UL72.

A segunda opção é a locação, que pode ocorrer em vários níveis, por exemplo:

- compartilhamento de servidor: um ou mais clientes dividem o processamento de uma máquina presente fisicamente em um ambiente seguro, normalmente com acesso e serviços limitados.
- aluguel da infra-estrutura física, como espacos em racks (com fornecimento de energia e conexão inclusas).
- *colocation*: locação de uma sala isolada dentro de um ambiente seguro. Este serviço é fornecido no Brasil por algumas empresas, como a Optiglobe (www.opti-

globe.com.br)

3.2 Soluções Gratuitas Analisadas

A lista que segue é o resultado de uma longa pesquisa na Internet, tanto através de ferramentas de pesquisa web como em newgroups específicos sobre o tema. Não foi fácil localizar estes produtos pois alem de existirem em pequeno número, não são divulgados e se perdem entre inúmeros outros projetos que somente fazem uso dos certificados digitais.

3.2.1 OpenSSL

Talvez o mais conhecido de todos os pacotes de criptografia, o OpenSSL surgiu da proposta da Netscape em criar um canal seguro (*Secure Socket Layer*, daí as letras SSL) entre navegadores e servidores web.

Esse software nada mais é que um implementação do SSL, atualmente na sua terceira versão. Uma das formas de operação deste protocolo faz uso dos certificados digitais e por este motivo foram incluídos no pacote funções e utilitários para a criação e uma gerência simples dos certificados. O utilitário principal para desempenhar as funções básicas de uma autoridade certificadora chama-se CA.pl e é implementado na linguagem PERL constituindo um *front-end* simples para as longas linhas de comando dos módulos do pacote OpenSSL.

É importante destacar que a forma de armazenamento e gerência dos certificados é mínima, utilizando arquivos no formato texto para os índices e guardando cada certificado em um arquivo separado, o que dificulta muito vários processos como a localização de um certificado e principalmente a manutenção desta base.

Esse sistema não pode ser considerado como um software para ICP pois não implementa muitas das funcionalidades necessárias ainda que seja largamente empregado como base para algumas implementações de sistemas para ICPs, como será visto.

3.2.2 OpenCA

O projeto do OpenCA é gerenciado usando um processo colaborativo e baseado no consenso dos participantes voluntários. O detalhe neste sistema é que ele é focado no usuário do software. Esse usuário é quem relata os problemas, envia sugestões e conduz os esforços da equipe de desenvolvimento.

O site reporta três projetos dos quais dois ainda estão em fase de ativação. O terceiro projeto é o que dá nome ao site e tem como objetivo desenvolver uma Autoridade Certificadora robusta, de código aberto e repleta de funcionalidades implementando os protocolos mais usados. O projeto OpenCA é baseado em vários outros projetos de código livre como OpenLDAP, Apache e principalmente o OpenSSL que é usado como base para a emissão dos certificados e todas as necessidades de criptografia desse servidor.

As principais vantagens são:

- Suporte para diversos bancos de dados (SGDB) através de módulos adaptadores

- Sistema de exportação de usuários e certificados, porém sem suporte ao backup da chave privada

- Divulgação de certificados por LDAP baseada no OpenLDAP

- Suporte à criptografia e geração dos certificados fornecidos pelo OpenSSL

Principais desvantagens:

- Suporte apenas aos sistema operacionais unix-like (não pode ser usado em plataforma Win32 devido às permissões dos usuários e grupos para segurança do sistema)

- Emprego da interface web para gerência dificulta a compreensão do funcionamento do sistema com um todo e a manutenção (criação e visualização) das políticas de certificação, que são as regras usadas para se autorizar a assinatura de um certificado.

- Dificuldade de modificação do código devida à grande fragmentação do

mesmo.

• Divulgação da situação dos certificados apenas por LCRs

3.2.3 IDX-PKI

O projeto IDX-PKI segue o padrão PKIX da IETF para infra-estrutura de chave pública. Ele foi desenvolvido pela companhia francesa IdealX que o publica e mantém sob licença GPL.

A IDX-PKI usa Perl, PHP, C, shellscripts, bem como alguns utilitarios GNU, e já está pronta para o uso diário, apesar de continuar sendo aperfeiçoado. Entre outras coisas, estão planejadas a capacidade de comunicação segura entre diferentes ICPs e mais um nível de abstração, que tornará possível escolher o repositório, seja banco de dados, LDAP ou sistema de arquivos.

As principais vantagens são:

• Grupo de desenvolvimento bastante ativo e focado nas funcionalidades;

• Apoio financeiro de uma empresa, que faz com que o grupo de trabalho permaneça coeso e focado

• Suporte às empresas que desejarem adotar o produto

• Adoção da licensa GNU

• Flexibilidade na modificação do código

• Armazenamento dos certificados usando LDAP ou banco de dados SQL

• Divulgação dos certificados usando LCRs ou OCSP

As desvantagens são:

• Modularidade excessiva, resultado da característica do grupo de trabalho, gerando muitos e pequenos módulos em linguagens diversas, resolvendo pequenos

problemas mas dificultando a expansão e a troca de componentes;

• Suporte apenas aos sistema operacionais *unix-like* (devido ao uso de shellscripts e utilitários da GNU)

• Dificuldade de instalação devida à dependência de módulos do PERL

• Falta de documentação

3.2.4 NewPKI

NewPKI é um projeto de desenvolvido nas horas vagas por uma única pessoa, Frédéric Giudicelli. Ele é baseado na API de baixo nível do OpenSSL e todos os dados são gerenciados através de um banco de dados, o que, segundo o autor provê, maior flexibilidade que o gerenciamento feito pelo OpenSSL, proporcionando, entre outras coisas, um forma simples de pesquisa.

Até o momento apenas o SGDB MySQL é suportado, porém devido à camada de abstração utilizada para acessar a base da dados, virtualmente qualquer outro banco de dados poderá ser empregado. A Implementação é feita em C++ o que possibilita a portabilidade do sistema para outras plataformas.

As vantagens são:

• Gerenciamento de múltiplas AC em um único servidor.

• Suporte ao controle de políticas de certificação.

• Revogação de certificados, geração e publicação via LCR ou OCSP.

• Visualização do status do processo de certificação via web e notificações por email.

• Linguagem única no desenvolvimento

Desvantagens da solução

• Ausência de um grupo de desenvolvimento. Uma única pessoa dita o cami-

nho a ser seguido

• O projeto possui um desenvolvimento lento pois o autor e desenvolvedor não se dedica a ele integralmente.

• O armazenamento não pode ser feito numa base LDAP

3.2.5 Características das Soluções OpenSouce

Esta seção, semelhante à existente para as soluções proprietárias, resume as principais características das soluções *OpenSource*.

É importante ressaltar que algumas tabelas de características foram omitidas por não existirem em nenhuma das soluções desta classe e do mesmo modo, outras tabelas foram incluídas mostrando características que são apenas pertinentes às soluções *OpenSource*.

Tabela 3.10: Sistema Operacional e Repositórios suportados

Produto	Compaq Tru64	Microsoft Win NT4	Microsoft Win 2K	Solaris 7 e 8	HP UX 11.0	AIX 4.3.3+	Linux
Open-CA	Não	Não	Não	LDAP	Não	Não	LDAP
IDX-PKI	LDAP/ MySQL	Não	Não	LDAP/ MySQL	LDAP/ MySQL	LDAP/ MySQL	LDAP/ MySQL
NewPKI	Não	MySQL	MySQL	MySQL	Não	Não	MySQL

Tabela 3.11: Métodos de Revogação de Certificados

Produto	CRL	OCSP	CRT[a]	CRL Distribution Point
Open-CA	Sim	Não	Não	Não
IDX-PKI	Sim	Sim	Não	Não
NewPKI	Sim	Sim	Não	Não

a. CRT - Certificate Revocation Tree

Tabela 3.12: Segurança das Comunicações

Produto	Open-CA	IDX-PKI	NewPKI
Comunicação com Clientes	Somente via web, usando SSL	Somente via web, usando SSL	Somente via web, usando SSL
Comunicação AC-RA	N/A	N/A	Via SSL
Hard e Soft para guarda dos certificados?	Sim, os suportados pela Engine do OpenSSL	Sim, os suportados pela Engine do OpenSSL	Sim, os suportados pela Engine do OpenSSL
Chave da AC protegida por Hardware?	Não	Não	Sim

Tabela 3.13: Característica da Autoridade Certificadora

Característica	Open-CA	IDX-PKI	NewPKI
Replicação do Repositório	Não	Sim	Não
Customização do Certificado por Cliente	Não	Não	Não
Validações de Terceiros	N/A	N/A	N/A
Funções para Manutenção do Repositório (checagem de integridade, encriptação, etc)	Não	Não	Não
Operações sobre Volumes (relatórios, monitoramento e etc)	Não	Não	Não
Extensões de certificados Suportadas (S/MIME, PKIX, IPSec, SSL)	Apenas com configuração manual	Apenas com configuração manual	Apenas com configuração manual
API para integração com outros Sistemas	Não	Não	Não
Repositório encriptado	Não	Não	Não
Número Máximo de Usuários por AC	Indeterminado	Indeterminado	Indeterminado
Número Máximo de ACs por Servidor	Indeterminado	Indeterminado	Indeterminado

Tabela 3.14: Característica do Projeto

Produto	Open-CA	IDX-PKI	NewPKI
Documentação	Deficiente	Deficiente	Satisfatória, ainda que incompleta
Grupo	Grupo ativo e composto na totalidade por voluntários	Grupo coeso e apoiado por uma empresa	Não existe um grupo de desenvolvimento, é o projeto de uma única pessoa.
Linguagem empregada	Várias, as principais são C e Perl	Várias, as principais: C, Perl e PHP	C/C++

3.2.6 Solução Selecionada: NewPKI

Para efeito da realização da implementação, todas as soluções anteriores foram estudadas, instaladas e avaliadas (o Apêndice V documenta o processo empregado na instalação da NewPKI). Os pontos considerados foram:

- Documentação
- Funcionalidades
- Flexibilidade de alteração
- Aderência aos padrões existentes

A complexidade da instalação, que envolve todo o processo de obtenção dos componentes necessários tanto para a operação (por exemplo servidor web ou de banco de dados) quanto para a compilação dos fontes (bibliotecas externas) e a instalação e configuração dos mesmos, não foi levada em consideração pois algumas distribuições do GNU/Linux já instalam e configuram automaticamente estes componentes ou, quando isso não é feito, fornecem ferramentas administrativas que facilitam este processo. Do mesmo modo, não foi considerada a questão de desempenho pois cada ajuste para um ambiente específico conduziria a um resultado diferente.

A instalação e os testes mostraram que:

• o OpenSSL assim como o OpenCA não implementam ICPs completas, ainda que caminhem nesta direção

• a IDX-PKI mostrou-se uma boa solução pelas funcionalidades que implementa e pela velocidade de desenvolvimento do projeto, mas seu código é muito segmentado e por isso de difícil manipulação.

A solução que mais se destacou com relação aos pontos considerados foi a NewPKI. A documentação existente ainda que resumida é bem satisfatória. A estrutura proposta nesta solução é bem simples e funcional e possui a grande vantagem de empregar um banco de dados conhecido (MySQL) para o armazenamento dos certificados e logs.

Todo o processo de comunicação do servidor com o módulo de gerência não é via HTTP, porém é seguro devido ao emprego de um canal cifrado usando SSL. Existe uma interface de gerência gráfica para Windows. Grande parte da operação segue os padrões, principalmente as partes mais críticas como o processo de requisição.

Existe uma interface web que é utilizada para a requisição dos certificados e é extremamente útil pois cria uma forma que possibilita ao requerente acompanhar o processo de certificação visualizando a condição da sua soliticação.

O código fonte, escrito em C++ e orientado a objetos, permite fácil compreensão e conta com uma divisão coerente das classes.

3.2.6.1 Funcionalidades Adicionadas

Analisando as características das soluções comerciais e confrontando com as necessidades da implementação de avaliação verificou-se que duas funcionalidades extras eram desejáveis, e não existiam na NewPKI, que são:

• Sistema de aviso de expiração do certificado com ajuste de antecedência, o que pode evitar que um usuário fique sem ter um certificado válido inadvertidamente.

• Sistema de backup da AC (incluindo configurações e a base de dados)

A primeira funcionalidade foi implementada utilizando um script, feito em PHP executado periodicamente pelo sistema que abriga a AC via entrada na tabela do crontab. O

segundo foi implementado utilizando a ferramenta MySQLCC, gratuita e mantida pela mesma equipe que desenvolve o MySQL. Essa solução foi escolhida devido ao grupo que a desenvolve conhecer profundamente o produto e pelo fato dela ter suporte a conexões seguras com o banco de dados usando SSL o que faz com que o procedimento de backup seja seguro.

3.2.6.2 Interface WEB

Visando a integração da NewPKI com o sistema do AirStrike foi utilizada uma interface WEB, escrita em PHP, que permite esta integração (vide Figura 3-2). Esta interface provê as funcionalidades de uma Autoridade Registradora e é baseada numa interface fornecida pelo desenvolvedor da NewPKI que na ocasião da obtenção não estava totalmente funcional.

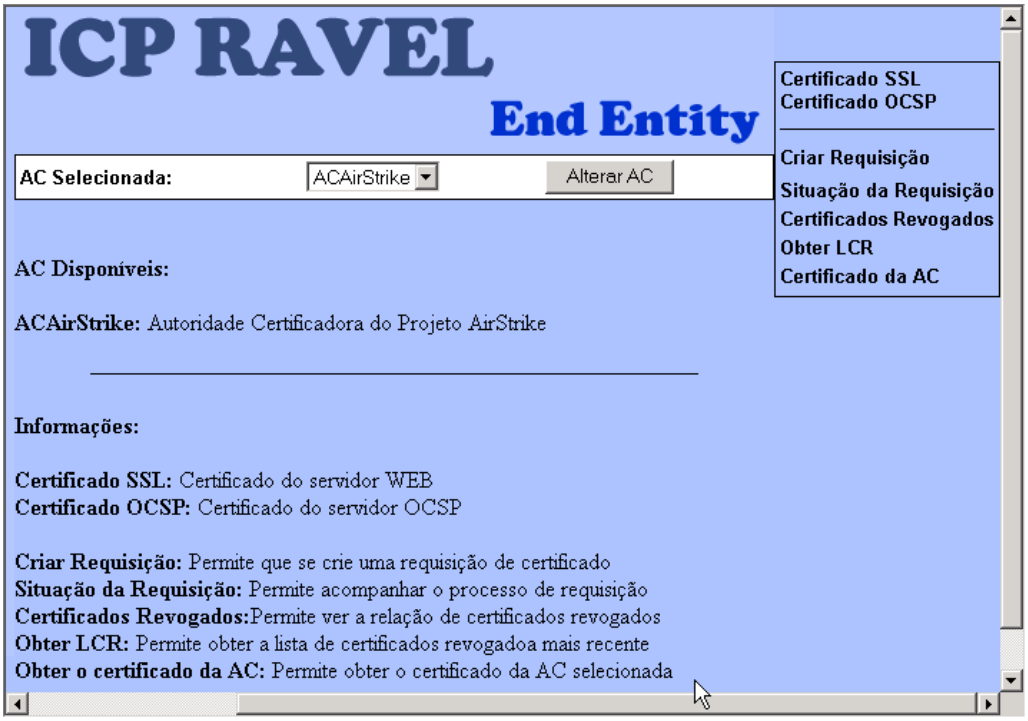


Figura 3-2. Tela inicial da interface web da NewPKI

Nessa tela inicial existe um menu lateral composto por 7 opções que são:

- (1)Certificado SSL: provê o mecanismo necessário para que o cliente possa obter certificado do servidor SSL utilizado nas conexões seguras.

- (2)Certificado OCSP: do mesmo modo, esta opção permite que o cliente

obtenha o certificado do OCSP *Responder* utilizado por todas as Autoridade Certificadora listadas.

• (3)Criar Requisição: esta opção é utilizada para efetuar a solicitação de certificados dos clientes (vide Figura 3-3 e Figura 3-4).

• (4)Situação da Requisição: após a solicitação, com esta opção, pode-se acompanhar o andamento do processo de emissão (vide Figura 3-5) e ao final (vide Figura 3-6) pode-se obter o certificado.

• (5)Certificados Revogados: apresenta uma relação de certificados revogados.

• (6)Obter LCR: permite ao cliente obter a lista de certificados revogados mais recente.

• (7)Certificado da AC: permite ao cliente obter o certificados da AC selecionada no menu central da tela.

.

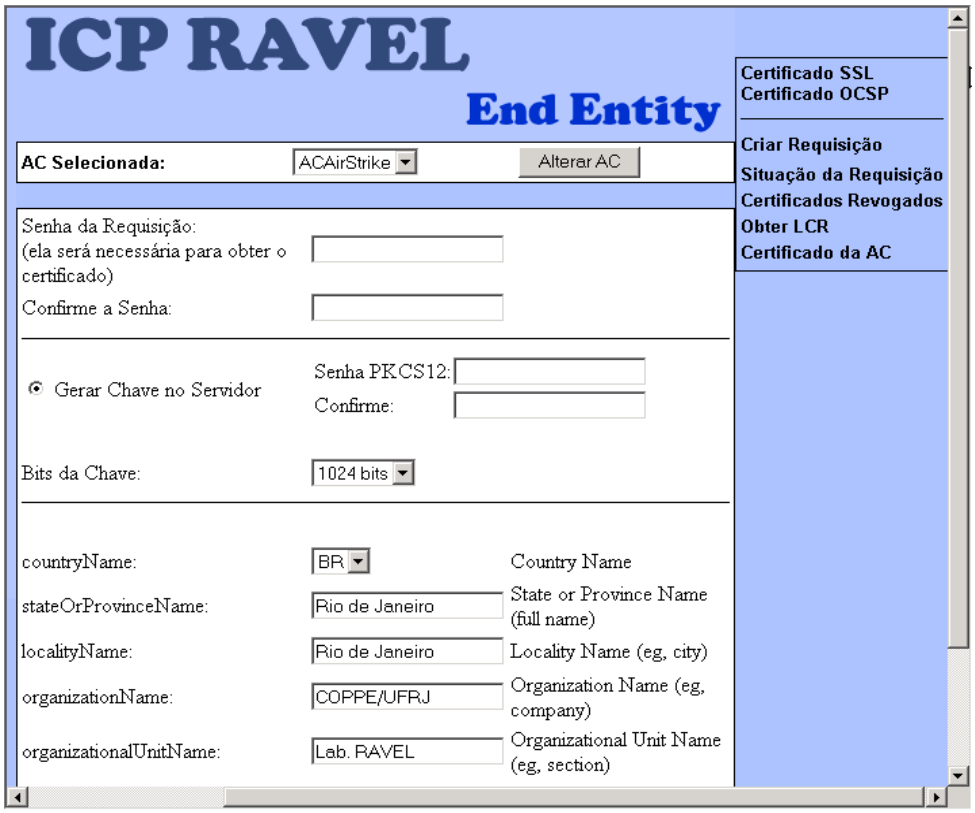


Figura 3-3. Formulário de requisição de certificado

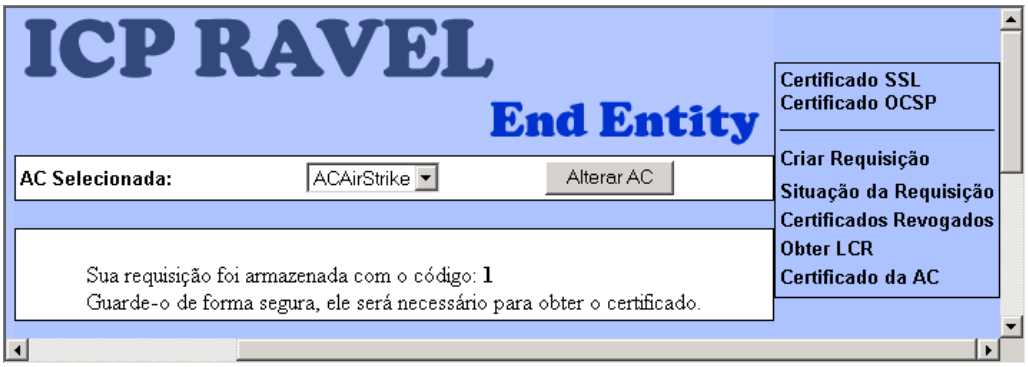


Figura 3-4. Tela de confirmação de solicitação

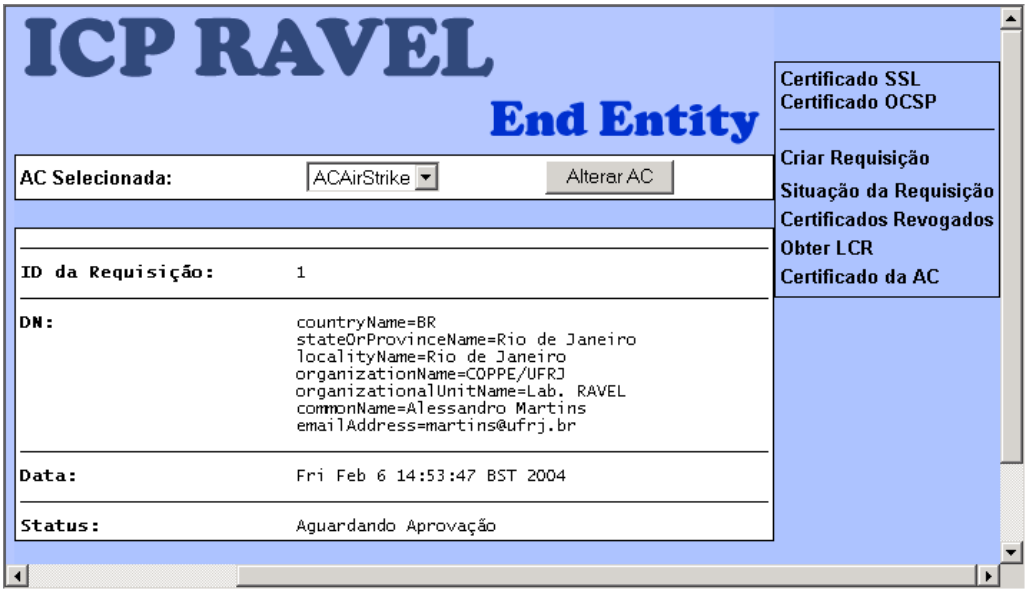


Figura 3-5. Informação sobre o andamento do processo de certificação

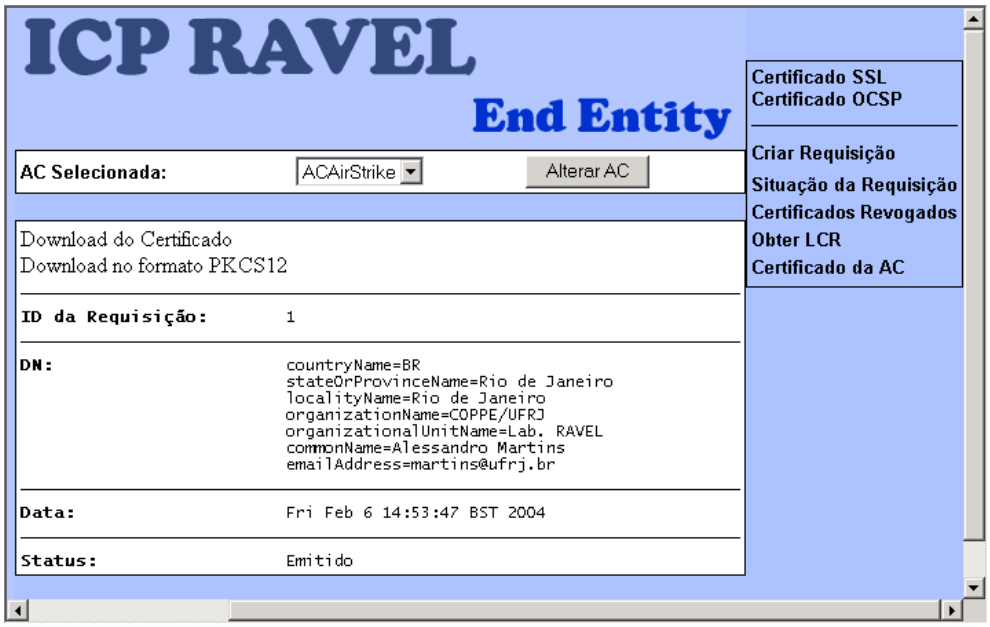


Figura 3-6. Tela final confirmando a emissão do certificado

3.2.6.3 Viabilidade do uso da Solução OpenSource

Existe atualmente uma tendência das empresas e principalmente de alguns governos de migrar do software proprietário para as versões livres. Como exemplos recentes temos a Venezuela[41] e o governo do Rio Gande do Sul o Metrô/SP[42] e a *Defense Information Systems Agency* (DISA), divisão de sistemas de informação do departamento de defesa norte-americano (DOD)[43] que estão deixando de usar a suite de aplicativos de código fechado, como o Microsoft Office, e passando para as versões livres e compatíveis como o StarOffice e o OpenOffice. A principal motivação ainda é o alto custo do sofware proprietário, porém o fato das versões gratuitas serem de código aberto traz novas razões, como por exemplo o ajuste segundo as necessidades, maior segurança, capacidade de ser auditado, entre outras.

Alguns pesquisadores advocam que o incentivo à criação e uso do software livre tem impactos benéficos sobre a questão social como relata Fábio Kon no relatório "O software aberto e a questão social"[40]. Richard Stallman, precursor deste movimento e notório defensor do software livre, em várias situações, como na palestra proferida no MIT por ocasião do *Communications Form* em 19 de abril de 2001 também aborda o assunto e se mostra favorável à idéia. Organismos como a *Free Software Fundation*

(FSF) e a Associação Brasileira de Software Livre (ABRASOL) têm auxiliado muito neste sentido, divulgando a filosofia, apoiando o desenvolvimento e a adoção.

Ainda é difícil determinar a penetração deste novo paradigma entre as diversas áreas do conhecimento o que torna complexo determinar quais são as motivações reais para essa mudança. Em geral dois pontos principais que parecem ser comuns à maioria dos ramos de negócios não centrados na tecnologia da informação, que são:

• Personalização e Adequação

Nas situações em que o código fonte é livre torna-se possível a alteração do mesmo gerando novas versões que personalizam o produto e/ou adequam o seu uso ou desempenho as reais necessidade. Qualquer necessidade futura pode ser novamente inserida, bastando em muitos casos apenas uma recompilação do código.

• Custo

Este ítem precisa ser considerado sob dois aspectos: primeiro o custo de obtenção e o segundo o custo de treinamento e/ou suporte.

O primeiro é uma questão importante, principalmente para empresas cuja natureza do negócio não seja tecnologia. Existem classes de aplicativos de uso geral, como editores de texto, planilhas e outros que podem ser obtidos gratuitamente e não precisam de terinamento e/ou suporte.

O segundo aspecto é fundamentalmente relacionado com os aplicativos que necessitam de treinamento e/ou suporte constante. Neste caso, em geral, o grande número de usuários garante o suporte necessário fazendo com que o aprendizado seja incremental, conforme a necessidade.

Existe um terceiro motivo muito ligado a outra classe de empresas, as centradas na tecnologia da informação, que precisam levar em consideração outras questões como: correção de falhas, continuidade do desenvolvimento, segurança e confiança.

Com o código fonte disponível pode-se fazer o que se desejar com ele, principalmente correções. Em geral, os softwares livres possuem muitos interessados e desenvolvedores que por razões diversas dispendem esforços não só na expansão dos recursos mas também na correção das falhas. Isso ocorre em muitos níveis, mas em geral torna-se um

esforço mundial fazendo com que o grande número de usuários acelere a detecção das falhas e conseqüentemente a correção destas.

A segurança e a confiança nesses programas é ultimamente o ponto mais atacado pelas empresas que produzem programas com código fechado.

Recentes vunerabilidades encontradas no servidor Apache (o mais usado segundo a pesquisa da E-Soft) e no OpenSSL levaram a revista eletrônica eWeek a avaliar a segurança dos programas de código aberto numa matéria entitulada "*Open Source: A False Sense of Security?*"[44]. Citando a opinião de várias partes com experiências tanto em software livre quanto proprietário, o consenso mostrou que o primeiro não é automaticamente mais seguro, mas geralmente o modelo de desenvolvimento do software livre[45] possibilita que as falhas sejam rapidamente solucionadas, o que aumenta muito a segurança.

Um relatório técnico entitulado "*Two Case Studies of Open Source Software Development: Apache and Mozilla*"[46] examina a argumentação de que o método de desenvolvimento do software de código aberto é comparável, senão melhor, que o método de desenvolvimento tradicional comercial. O autor formula várias hipóteses analisando dados do projeto do Apache e do Mozilla e conclui, com boas expectativas, que um processo de desenvolvimento híbrido será o adotado no futuro.

Um dos argumentos empregado pelas empresas de software fechado é que o custo e a segurança dos seus sistemas se baseia na produção própria do código, o que nem sempre pode ser comprovado ou é verdade. O fato é que em muitos casos o software livre é copiado e agregado a outros sem que isso seja divulgado. Como resultado, muitos de nós já usamos desavisadamente software livre (furtado) sem saber. Dois fortes exemplos comprovam esta afirmação:

1o. uma falha no código do FreeBSD que se encarrega de remontar os pacotes TCP/IP foi detectada e para surpresa dos pesquisadores a mesma falha foi encontrada numa versão do Windows, mostrando ao menos que a mesma idéia foi usada mas alguns garantem que a semelhança é tão grande que pode-se considerar que o mesmo código foi utilizado.

2o. semelhantemente, outra falha, agora no OpenSSL também foi encontrada no pacote criptográfico da RSA Security o RSABsafe. A diferença é que a solução para o primeiro foi quase imediata.

Estes dois casos derrubam o principal argumento das empresas de software fechado e estas não conseguem refutar outra grande vantagem do software aberto: a capacidade de ser auditado.

Capítulo

4

Proposta de Implementação do Protocolo OCSP

Este capítulo descreve o procedimento e os resultados obtidos durante o processo de desenvolvimento do cliente e do servidor OCSP a serem integrados com a solução de ICP selecionada anteriormente.

Em resumo, foram realizadas 2 fases: na primeira (seção 4.1) foi feita uma pesquisa visando obter informações e exemplos que pudessem auxiliar no processo de compreensão e desenvolvimento do programa. Na segunda (seção 4.2), foi realizada a implementação dos dois aplicativos que devido à natureza experimental, foram validados num último procedimento.

4.1 Estudo Preliminar

Esta primeira fase se caracteriza pela pesquisa e obtenção do material necessário à fase seguinte, de implementação.

Notadamente, poucas informações foram encontras sobre o funcionamento do protocolo OCSP e ainda menos sobre sua implementação. Numa pesquisa em sites nacionais, este fato fica ainda mais claro, apenas 111 resultados foram reportados, segundo pesquisa feita com a ferramenta Google em 21/01/2004 com a palavra OCSP, o que demonstra o estágio ainda embrionário de estudo e uso deste protocolo.

Algumas empresas que desenvolvem *plugins* para os principais produtos de infra-estrutura de chaves públicas fornecem informações sobre seus produtos que ajudam a compreender melhor a RFC 2560, que formaliza o protocolo. Dentre elas, a Vali-

cert[5] se destaca pois, em parceria com o Openvalidation[6], fornecem a única referência pública para implementadores desse protocolo.

Uma segunda fonte de informações foi obtida do Laboratório de Segurança Computacional da Escola Politécnica de Torino[7] (TORSEC). Este laboratório fornece além de informações, a única implementação compilada (para win32 e linux, os fontes não estão disponíveis) de um cliente e um servidor OCSP além de um conjunto de certificados para testes.

Para estudar a operação das implementações, foi utilizado o cliente TORSEC e três servidores: o também fornecido pelo TORSEC, o acessível para testes do OpenValidation e o incluso na solução selecionada. Os navegadores web com suporte ao OCSP não foram utilizados como clientes pois não disponibilizam as mensagens de erro e nem fornecem acesso fácil aos certificados de teste.

O resultado dos testes é apresentado na tabela que segue.

Cliente	Certificados	Servidor	Observações
TORSEC executado localmente em ambiente win32	fornecidos junto com o pacote de teste do TORSEC	TORSEC executado localmente em ambiente win32	As instruções de teste fornecidas funcionaram sem erro.
TORSEC executado localmente em ambiente win32	fornecidos junto com o pacote de teste do TORSEC	NewPKI	variações nos parâmetros do cliente levaram a resultados diferentes, apontando para problemas de implementação em uma das partes.
TORSEC executado localmente em ambiente win32	disponíveis no site do openvalidation	remoto no endereço ocsp.openvalidation.org	Algumas respostas não foram compreendidas corretamente pelo cliente.

Tabela 4-1. Resultado dos testes do cliente TORSEC

Com as informações obtidas, observou-se que existem duas formas de operacionalizar (por parte do cliente) a verificação de certificados usando OCSP. A primeira obtém do próprio certificado a ser verificado as informações necessárias, ou seja, a localização do OCSP *Responder* a ser usado. O procedimento é semelhante ao empregado para localizar a LCR associada a um certificado, diferindo apenas no objeto que carrega

5. www.valicert.net
6. www.openvalidation.org
7. http://security.polito.it/tools/ocsp/

esta informação. A definição do campo que armazena essa informação é feita na RFC 3260 (que atualiza a RFC 2459 mencionada na RFC 2560 com sendo a fonte das informações). No segundo procedimento o cliente ignora, caso exista, esta informação e utiliza um endereço fornecido pelo usuário. Este segundo procedimento possibilita que certificados gerados sem a localização de um *Responder* possam futuramente ser checados em um.

Após estes testes, foram conduzidos dois estudos para finalisar esta fase. O primeiro visando descrever o protocolo e o segundo com o objetivo de verificar a melhor forma de implementação. Ambos são apresentados em datalhes nas seções seguintes.

4.1.1 Descrição do Protocolo OCSP

O protoloco OCSP opera sobre no modo clássico de cliente-servidor. Toda a operação ocorre em apenas duas mensagens. Pelo lado do cliente é gerada uma requisição que é enviada, sem a necessidade de um canal seguro, para o servidor. Este por sua vez, retorna uma resposta também por um canal não necessariamente seguro. A utilização de um canal seguro não é necessária para garantir a autenticidade e integridade da requisição e da resposta, pois ambas são certificados digitais que podem estar assinados (a requisição, segundo o protocolo, não é obrigatória que esteja assinada).

Uma questão importante é sobre quem pode ser o servidor durante este processo. Como será visto, este protocolo estabelece apenas os formatos e os conteúdos das mensagens. A fonte de onde as informações necessárias para gerar a resposta não é estabelecida. Isso quer dizer que o servidor precisa: (a) ter acesso à base de informações, caso ele não seja o gerador da mesma, e (b) gozar da confiança do cliente. Neste cenário, pode ser um servidor:

(1) a própria AC que gerou o certificado

(2) uma AC especialmente delegada para tal fim, pela AC que gerou o certificado

(3) uma AC qualquer, mesmo sem vínculo com a AC que gerou o certificado, mas na qual o cliente confie.

Originalmente, o protocolo servia-se das informações existentes nas LCRs. Essa íntima relação é a motivadora dos campos temporais presentes nas respostas. Atualmente os certificados não são somente armazenados em diretórios e por isso as informações

podem ser obtidas de outras fontes, como banco de dados, adequadamente menos sensíveis à questão temporal, como são as LCRs.

A definição do OCSP também não amarra o protocolo de transporte, qualquer um pode ser utilizado, desde que compreendido pelas partes. O mais comum em funcionamento atualmente é o HTTP.

4.1.1.1 Conteúdo obrigatório da requisição do cliente

O protocolo requer que os seguintes campos estejam presentes na requisição:

- versão do protocolo
- um identificador do certificado a ser verificado

Outras informações, como extensões, que podem ser processadas ou não pelo *Responder* e a assinatura do requerente, podem ser incluídas na mensagem, mas não são obrigatórias.

4.1.1.2 Tipos de respostas do servidor

Durante seu uso, o servidor pode estar em dois estados, e para cada um desses existe um conjunto de respostas possíveis. Os dois estados são:

- Não operacional: este estado pode ocorrer nas seguintes condições:

- durante o processo de inicialização do servidor, ou
- durante seu processo de operação, quando algum problema interno (falha de operação ou exaustão dos recursos do sistema, por exemplo) ou externo (perda de conexão com alguma autoridade envolvida no processo) ocorrer.

Quando neste estado, o servidor somente retorna mensagens de erro não assinadas, que podem ser *internal error* ou *try later*.

- Operacional: neste estado o servidor está apto a processar as requisições. As respostas geradas neste estado, exceto as de erro, serão assinadas e compostas por:

- versão do protocolo
- nome do servidor
- informações temporais (nos campos *thisUpdate, nextUpdate e producedAt*)
- resposta para o certificado presente na resquisição do cliente

- extensões opcionais
- OID do algoritmos utilizado na assinatura
- assinatura computada sobre o *hash* da resposta (todos os campos já relacionados)

Para cada certificado checado, três estados podem ser informados: *good*, *revoked* e *unknown* e três mensagens de erro podem ser geradas: *signature required*, *mal formed request* ou *unauthorized*.

4.1.2 Caminhos para a implementação

Como nenhuma informação sobre como implementar o protocolo foi encontrada, foi necessário compreender melhor não a forma de funcionamento do protocolo, mas sim como gerar os objetos transportados pelo protocolo: os certificados digitais. Estes certificados possuem características semelhantes aos certificados comuns. Na verdade eles seguem o padrão PKIX, porém, campos e valores especiais são empregados tanto nos certificados usados nas requisições como nas respostas (vide Figura 4-1). Com essa constatação, foi feita uma pesquisa visando encontrar bibliotecas aptas a gerar certificados e executar as funções de criptografia assimétrica necessárias (basicamente todas as relacionadas com o algoritmos RSA). A lista presente no Apêndice II fornece apenas a relação das bibliotecas mais maduras encontradas. Para o desenvolvimento, a questão do grau de maturidade das bibliotecas foi crucial nas escolhas. Algumas bibliotecas jovens possuem APIs bem mais amigáveis (algumas até orientadas a objetos) e exemplos simples, material propício para a fase inicial de divulgação da biblioteca, mas não possuem documentação e nem um grupo estável de usuários, dois pontos fundamentais.

A primeira biblioteca estudada foi a SNACC, que somente manipula objetos ASN.1 tendo como saída código em C para uso em programas de geração de certificados. O uso dessa biblioteca implicaria numa abordagem extremamente em baixo nível, pois seria necessário implementar todas as descrições ASN.1 dos certificados PKIX a serem usados e após isso, seria necessário adotar outras bibliotecas para manipular os algoritmos, codificar os certificados e implementar o mecanismo de transporte, por isso esta abordagem não foi adotada dado o grau de complexidade e o fato de que algumas outras bibliotecas já possuem grande parte desse esforço implementado, como é o caso da Cryptlib e da OpenSSL.

Essas duas bibliotecas foram selecionadas, após uma avaliação junto com outras (vide Apêndice II) para as implementações devido ao seu largo emprego, maturidade, boa documentação e adequação ao propósito. Foram requisitos também para esta escolha o fato de ambas serem implementadas em C e serem portáveis.

4.2 Desenvolvimento e Validação

Ao concluir os testes com os aplicativos obtidos no início da fase preliminar, ficou claro que para uma correta implementação do cliente e do servidor seria antes necessário validar alguns conjuntos de certificados. Para isso foi necessário ler o conteúdo dos certificados fornecidos pelo TORSEC e pelo OpenValidation. Para este fim foram criados dois aplicativos auxiliares (cada um usando uma das bibliotecas selecionadas) que recebem como entrada um certificado e apresentam o seu conteúdo em formato texto. Este passo foi importante para comprender melhor os exemplos da documentação, o funcionamento e a forma de programação de cada uma das bibliotecas. Os dados dos certificados presentes no Apêndice III foram gerados usando a versão OpenSSL do aplicativo.

Foi através de um processo de comparação dos conteúdos dos certificados fornecidos com as informações da RFC 2459/3260 e da RFC 2560 que foram gerados os certificados de teste. Alguns certificados extras também foram obtidos e utilizados como fonte de informação, como o do Banco do Brasil e da Tumbleweed (vide Apêndice III), apenas para verificar a correta compreensão da RFC 3260 no que tange os objetos necessários no certificado.

O conjunto de teste é composto por:

- um certificado válido a ser verificado

- um certificado revogado a ser verificado

- um certificado da AC e do *Responder* com suas respectivas chaves privadas e senhas de acesso (necessárias para poder assinar a resposta)

- um certificado e sua correspondente chave privada com a senha de acesso do cliente que irá fazer a solicitação (necessário para que a requisição possa ser assinada)

Com este conjunto de certificados foram refeitos os testes anteriores. A primeira sequência de teste não obteve sucesso pois a sintaxe do OID responsável por informar a

localização do OCSP *Responder* estava errada. O formato correto exige a presença do protocolo de transporte, não apenas do endereço do *Responder*. Corrigida esta falha, os resultados encontrados foram os mesmos obtidos anteriormente, indicando que os certificados gerados são funcionalmente semelhantes aos anteriores. Este procedimento encerrou a fase de validação dos certificados. De posse destes certificados e com o auxílio dos servidores existente foram testados dois clientes desenvolvidos, cada um usando uma das bibliotecas selecionadas.

Ainda que estruturalmente diferentes, devido à diferença de abordagem de cada biblioteca, funcionalmente os clientes foram desenvolvidos seguindo o esquema da Figura 4.1.

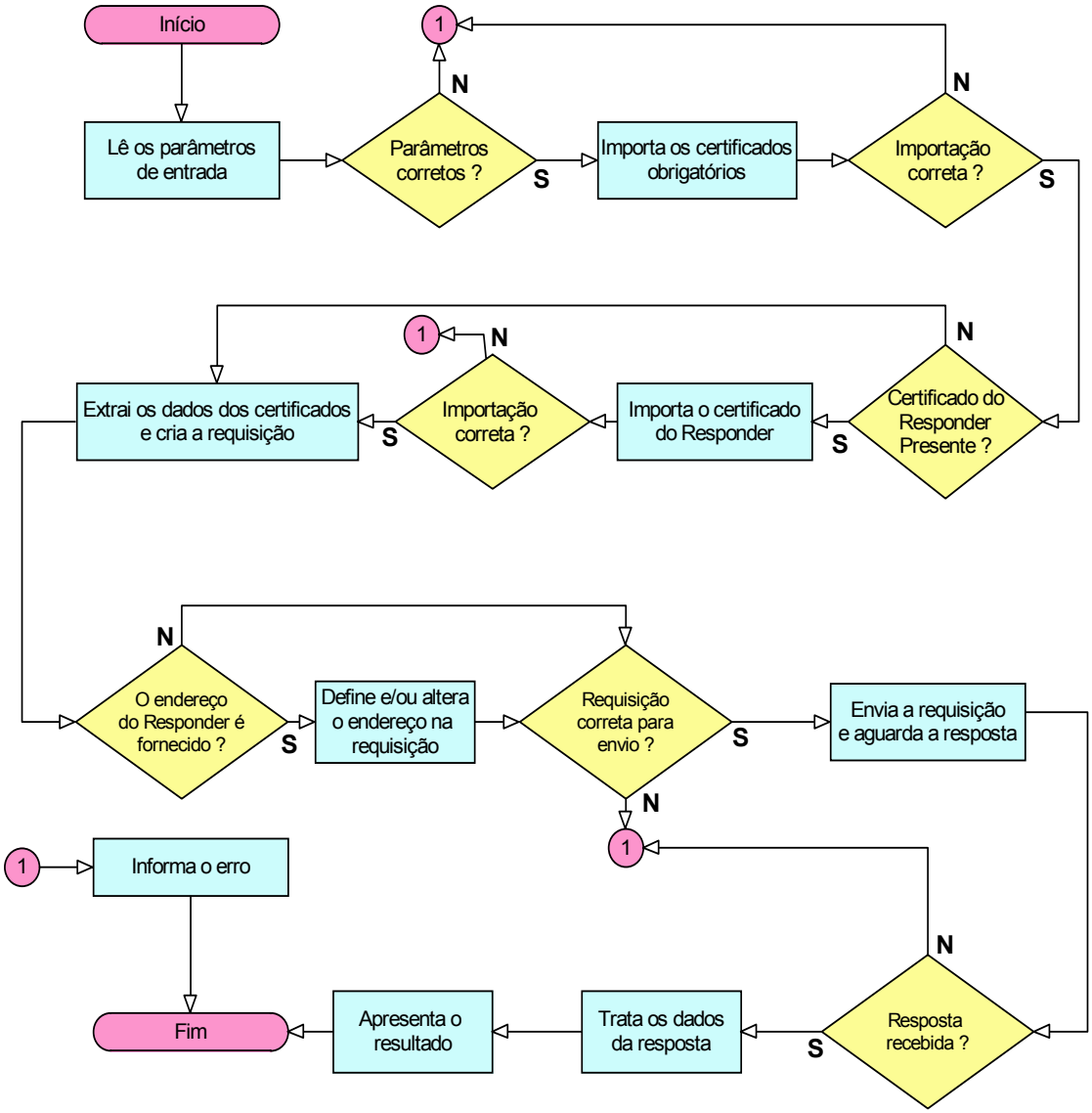


Figura 4-1. Esquema funcional do cliente OCSP

Nestes esquema, ambos recebem como parâmetros de entrada:

- o certificado a ser verificado

- o certificado da AC que emitiu o certificado

- opcionalmente o certificado do *Responder* caso este não seja a AC que emitiu o certificado

- opcionalmente o endereço do *Responder*. Se um endereço for fornecido o existente no certificado será ignorado, caso nenhum seja fornecido e não exista um no certificado este erro será informado. O certificado a ser verificado também será usado para assinar a requisição apenas para reduzir a quantidade de certificados em uso e simplificar o procedimento de teste.

Estes clientes foram testados tendo como servidor o fornecido pelo TORSEC e o existente no NewPKI. O servidor do OpenValidation não pode ser usado pois não é possível tranferir para eles os certificados gerados e necessários na criação da resposta gerada pelo *Responder*. Os resultados foram os seguintes:

Cliente	Certificado	Servidor	Observações
cliente desenvolvido usando OpenSSL	conjunto de teste	TORSEC executado localmente em ambiente win32	resultado inconsistente na rotina OCSP_check_validity da biblioteca OpenSSL
cliente desenvolvido usando OpenSSL	conjunto de teste	NewPKI	ídem ao anterior
cliente desenvolvido usando Cryptlib	conjunto de teste	TORSEC executado localmente em ambiente win32	devido aos controles de segurança da biblioteca cryptlib o cliente apresentou um problema na leitura do formato dos certificados, que foi corrigido alterando-se o nível de compatibilidade desta.
cliente desenvolvido usando Cryptlib	conjunto de teste	NewPKI	ídem ao anterior

Tabela 4-2. Resultados dos testes executados com clientes implementados

Validados os clientes e os conjuntos de certificados iniciou-se o desenvolvimento dos servidores.

Dada a semelhança do formato da requisição com o da resposta, o procedimento de desenvolvimento do servidor foi bem semelhante ao do cliente, seguindo o esquema da Figura 4.2. A diferença básica é que o servidor precisa permanecer ativo após cada

resposta gerada por ele e que as respostas válidas precisam ser assinadas e para isso torna-se obrigatório fornecer a chave privada e a senha de acesso a ela como parâmetros ao programa.

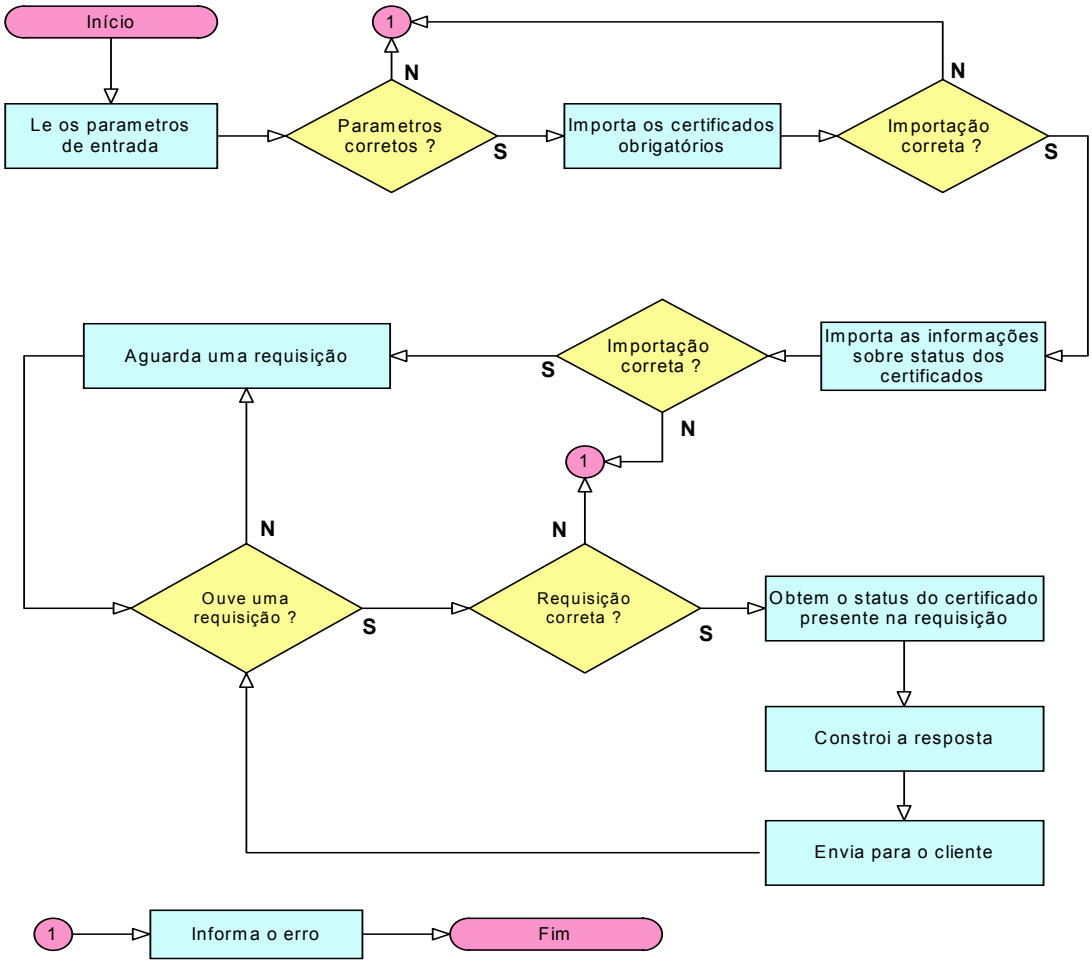


Figura 4-2. Esquema funcional do servidor OCSP

Os servidores desenvolvidos foram testados com os clientes e certificados já validados. Os resultados apresentados na tabela seguinte mostram que a operação do conjunto foi completada corretamente.

O resultado inconsistente obtido da rotina `OCSP_check_validity` da biblioteca OpenSSL está relacionado com a configuração do fuso horário do sistema operacional usado nos teste (Windows 2000). Esse erro não interfere na operação do protocolo de uma forma geral.

Cliente	Certificado	Servidor	Observações
cliente desenvolvido usando OpenSSL	conjunto de teste	servidor desenvolvido usando OpenSSL	resultado inconsistente na rotina OCSP_check_validity da biblioteca OpenSSL
cliente desenvolvido usando Cryptlib	conjunto de teste	servidor desenvolvido usando Cryptlib	todos os testes ocorrem sem erros
cliente desenvolvido usando OpenSSL	conjunto de teste	servidor desenvolvido usando Cryptlib	resultado inconsistente na rotina OCSP_check_validity da biblioteca OpenSSL
cliente desenvolvido usando Cryptlib	conjunto de teste	servidor desenvolvido usando OpenSSL	todos os testes ocorrem sem erros
TORSEC executado localmente em ambiente win32	conjunto de teste	servidor desenvolvido usando Cryptlib	as respostas foram obtidas corretamente
TORSEC executado localmente em ambiente win32	conjunto de teste	servidor desenvolvido usando OpenSSL	as respostas foram obtidas corretamente

Tabela 4-3. Resultado dos testes dos clientes e servidores desenvolvidos

Capítulo

5 Medições e Resultados

O objetivo deste capítulo é apresentar uma série de medições e simulações[8] visando gerar material suficiente para que seja possível tecer algumas comparações entre dois métodos de verificação de situação dos certificados: o protocolo OCSP e o mecanismo das LCRs. Essas comparações serão feitas em duas frentes: uma focada na utilização da largura de banda e outra no esforço computacional na geração e processamento do material.

5.1 - Largura de Banda

Para avaliar o consumo de largura de banda da rede foi estudado o comportamento das LCRs com relação à variação do tamanho das listas no tempo, que possui relação direta com a quantidade de certificados, como será visto.

Para comparar o impacto na rede da utilização tanto do protocolo OCSP quanto das LCRs num ambiente é necessário:

- caracterizar o tamanho das LCRs ao longo do tempo e para isso os parâmetros principais são: o número de identificadores de certificados revogados, a freqüência de revogação e o tempo de validade dos certificados e

- caracterizar o número de requisições ao longo do tempo dessas LCRs que têm como parâmetros principais além da freqüência de promulgação a freqüência das consultas.

Todos estes parâmetros têm íntima relação com o número de usuários e sua taxa de crescimento. Cobrir todos estes parâmetros com simulações poderia conduzir a resultados irreais pois ainda não existe nenhum modelo que os leve em consideração numa caracte-

8. Todas as simulações foram executadas num Pentim III 1GHz com 128Mb e disco IDE-33 de 5200rpm usando Windows 2000 em iguais condições, ou seja, sempre após uma inicialização do sistema.

rização de ambientes. Estudando mais a fundo a questão, percebeu-se uma forma de modelar o comportamento do tamanho das LCRs no tempo, frente a um de seus parâmetros principais, em condições de uso reais. Isto foi feito obtendo com certa freqüência as LCRs promulgadas por grandes certificadoras e acompanhando a sua evolução.

As métricas consideradas na modelagem das LCRs foram:

- o tamanho do arquivo digital que abriga a LCR, medido em bytes,
- a quantidade de indicadores de certificados revogados presentes no arquivo da LCR,
- a quantidade de dígitos utilizada para formar os identificadores dos certificados revogados.

Essas métricas, mesmo em número reduzido, mostraram que mesmo sem o conhecimento de parâmetros que só são possíveis de serem obtidos de dentro dos processos de revogação, e por conseguinte, dentro das aplicações que os executam, como a freqüência das revogações, é possível modelar as LCRs e com isso estimar o seu tamanho para certas classes de certificadoras. Os resultados deste processo são apresentados na seção seguinte.

5.1.1 Tamanho das LCRs de ACs Comerciais

Inicialmente foram pesquisadas algumas autoridades certificadoras que pudessem fornecer material para este estudo. As pesquisadas foram a Verisign, a RSA Security e a Unicert do Brasil. Pela abrangência mundial, a Verisign foi selecionada para ser analisada, pois possui em seu endereço de acesso as LCRs mais de 50 listas associadas a diversas políticas de certificação e perfis de usuários, cobrindo uma ampla faixa de possibilidades, como demonstram os gráficos. As outras empresas possuem uma quantidade não significativa de listas, por isso foram descartadas.

Para levantar as métricas definidas, foram obtidas por download automático (os detalhes do processo são apresentados no Apêndice III) todos os dias durante um mês todas as listas da Verisign. Após ter uma cópia local das listas, estas foram processadas por um aplicativo especialmente desenvolvido para este fim que extrai 3 parâmetros: o

tamanho do arquivo (em bytes), o número de identificadores de certificados na LCR e a versão do formato da lista.

De posse desses valores foram geradas uma série de curvas e a observação das mesmas levou a definição de 5 perfis distintos, que foram nomeados da seguinte forma:

• constante: caracterizado pela estabilidade no tamanho da lista ao longo do tempo (vide Figura 5-1). Esta lista deve provavelmente estar associada a uma certificadora de alto nível hierárquico ou de aplicação extremamente particular e segura onde as revogações praticamente não ocorrem. Formam o maior grupo das LCRs verificadas, com 80% das ocorrências.

• crescente: caracterizado pela evolução monotônica do tamanho da lista (vide Figura 5-2), representa possivelmente uma certificadora final, associada diretamente aos usuários onde a cada nova emissão o número de certificados que entram na lista (recém revogados) é sempre maior do que os que saem dela (os expirados). Nessa classe, verificou-se que as listas são emitidas diariamente, ainda que sua validade seja de 10 ou 14 dias. Ocorrem em pequena quantidade, apenas 8% do total das LCRs verificadas.

• decrescente: com característica oposta a anterior (vide Figura 5-3), representa possivelmente uma certificadora em fase de encerramento de suas operações, aguardando apenas que todos os certificados emitidos por ela expirem. Este procedimento pode ser utilizado para executar uma mudança de política, por exemplo. Também ocorrem em pequeno número, apenas 4% do total.

• degrau: caracterizada pela mudança brusca no tamanho da lista (vide Figura 5-4), pode ser uma variação da classe constante por ocasião de um revogação exporádica ou um degeneração da classe crescente, onde ocorrem poucas revogações e por isso a lista não é freqüentemente atualizada. Ocorre em aproximadamente 4% das listas verificadas.

• tangente: caracterizada pela semelhança com a função matemática. Pode ser vista como uma variação do perfil crescente onde os certificados recem revogados são constantemente adicionados à lista e os expirados são removidos apenas

por ocasião das promulgações. Ocorre em aproximadamente de 4% das listas verificadas.

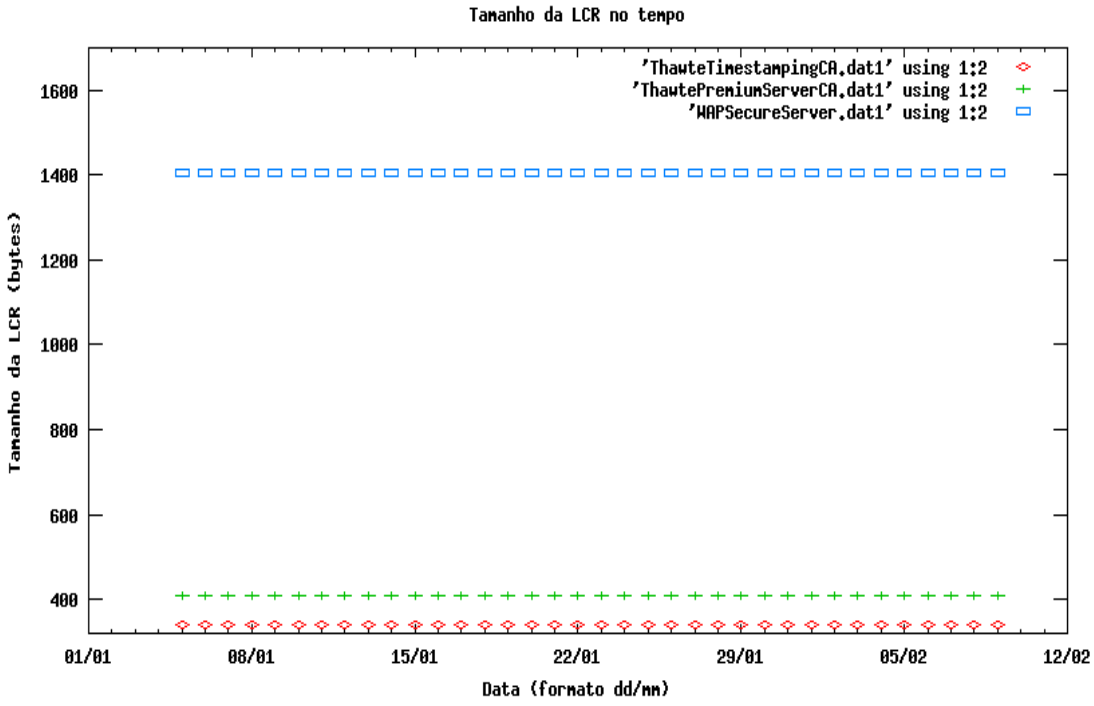


Figura 5-1. Exemplo de comportamento da Classe Constante de LCRs

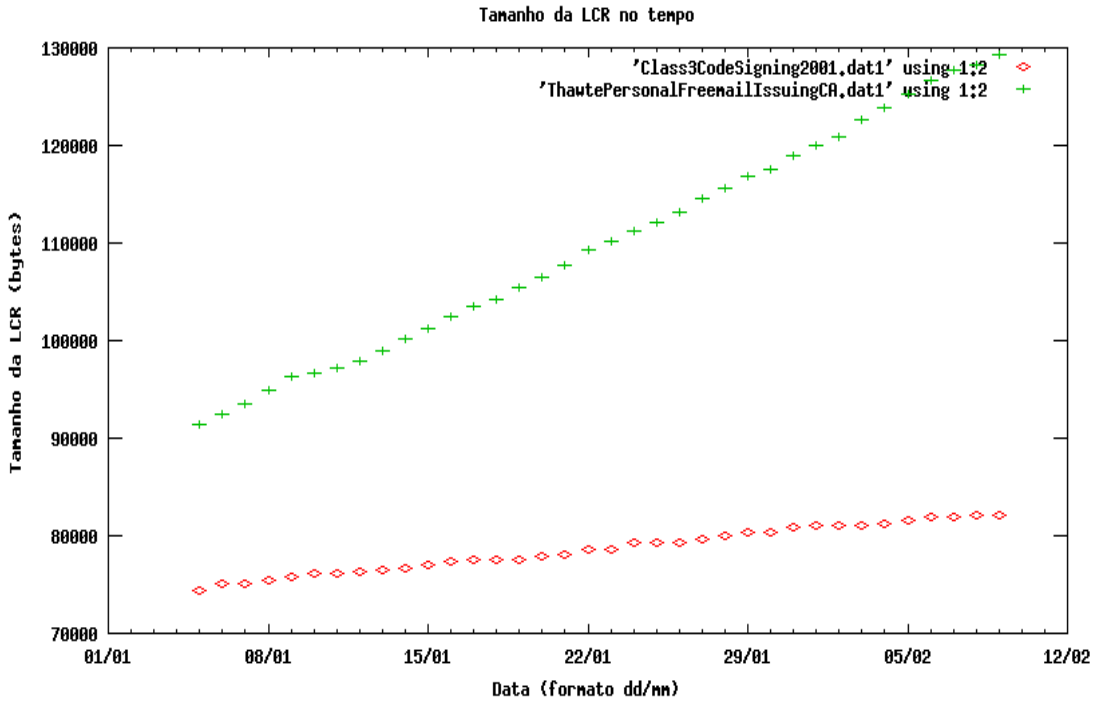


Figura 5-2. Exemplo de comportamento da Classe Crescente de LCRs

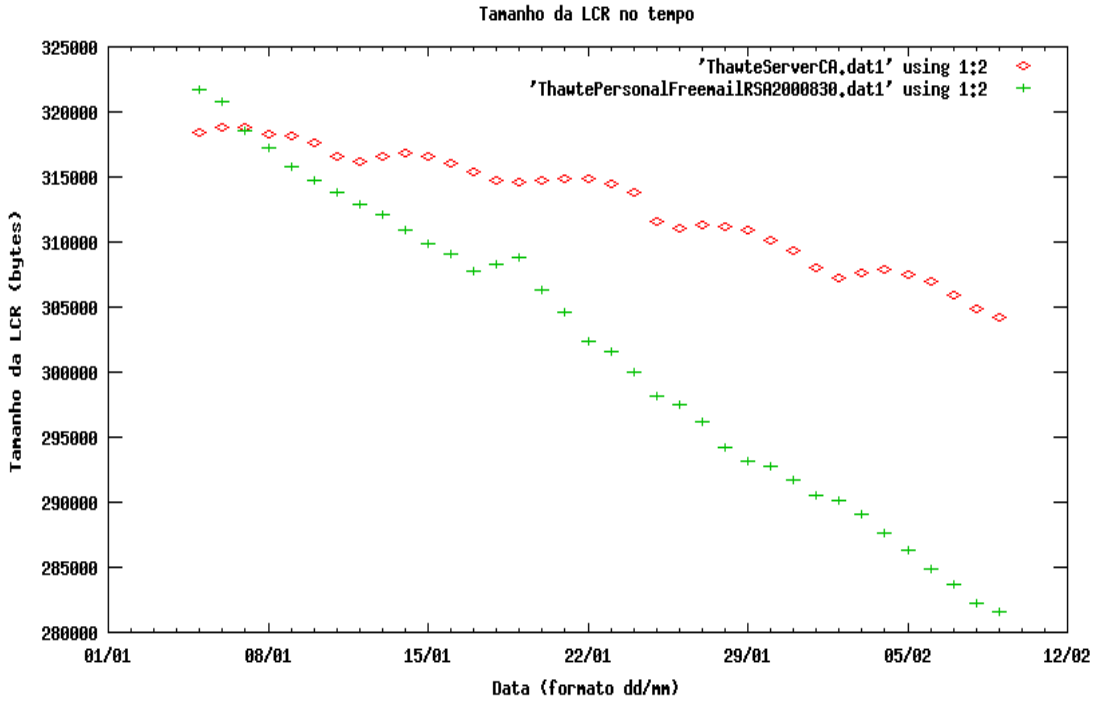


Figura 5-3. Comportamento da Classe Decrescente de LCRs

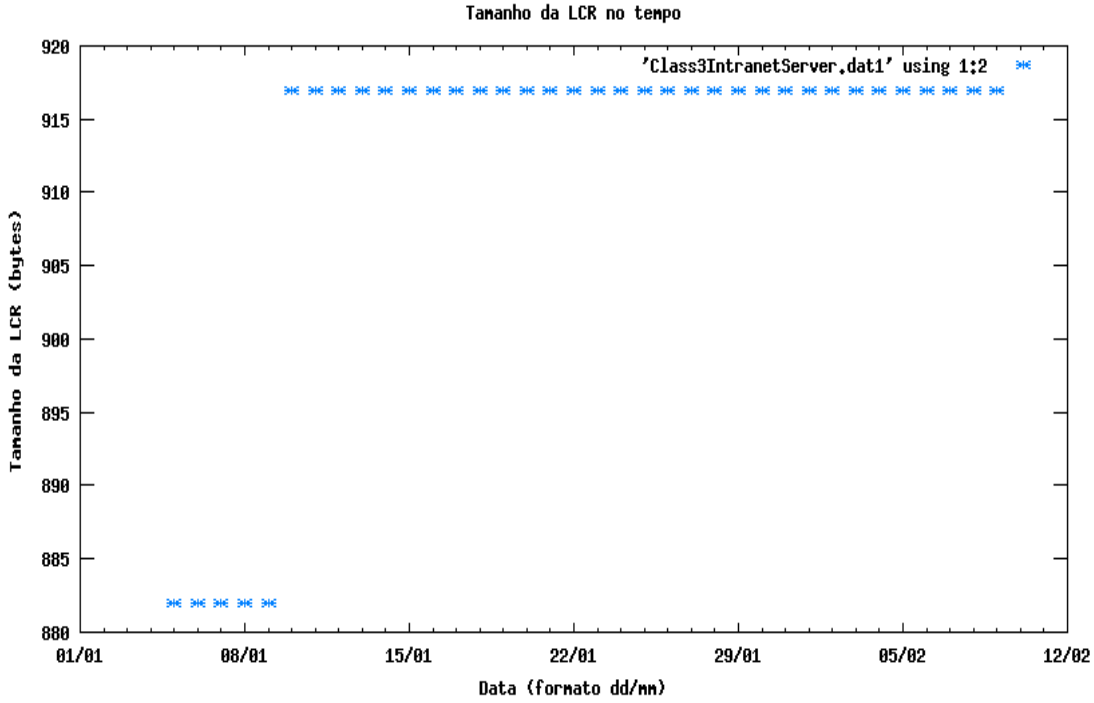


Figura 5-4. Comportamento da Classe Degrau de LCRs

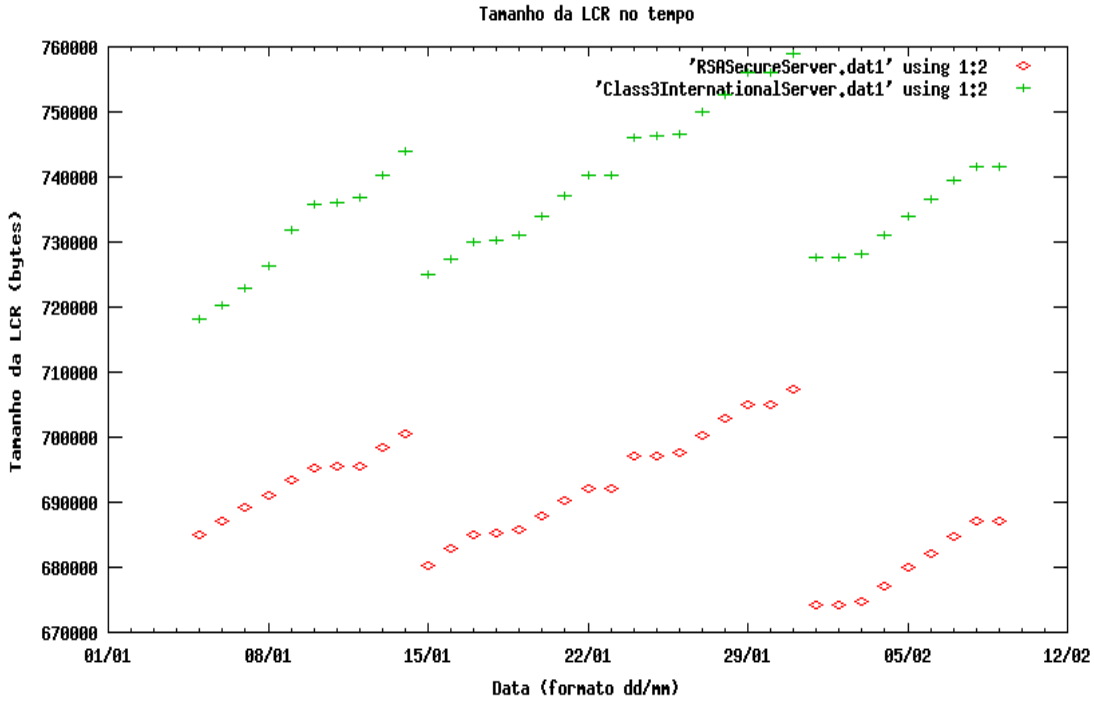


Figura 5-5. Comportamento da Classe Tangente de LCRs

Além dos perfis detectados, verificou-se observando o valor obtido para a versão das listas das da Verisign que todas elas adotam a priveira versão do formato de LCRs.

Na tentativa de obter LCRs de acordo com a segunda versão da lista, o procedimento de download foi também aplicado às listas da RSA Security e da Unicert do Brasil. Com isso foi possível verificar que ambas usam a segunda versão da lista, que mesmo não tendo entrado em amplo uso, possui características importantes, apresentadas na seção 5.3.

Os gráficos que seguem apresentam os valores obtidos das listas dessas duas certificadoras.

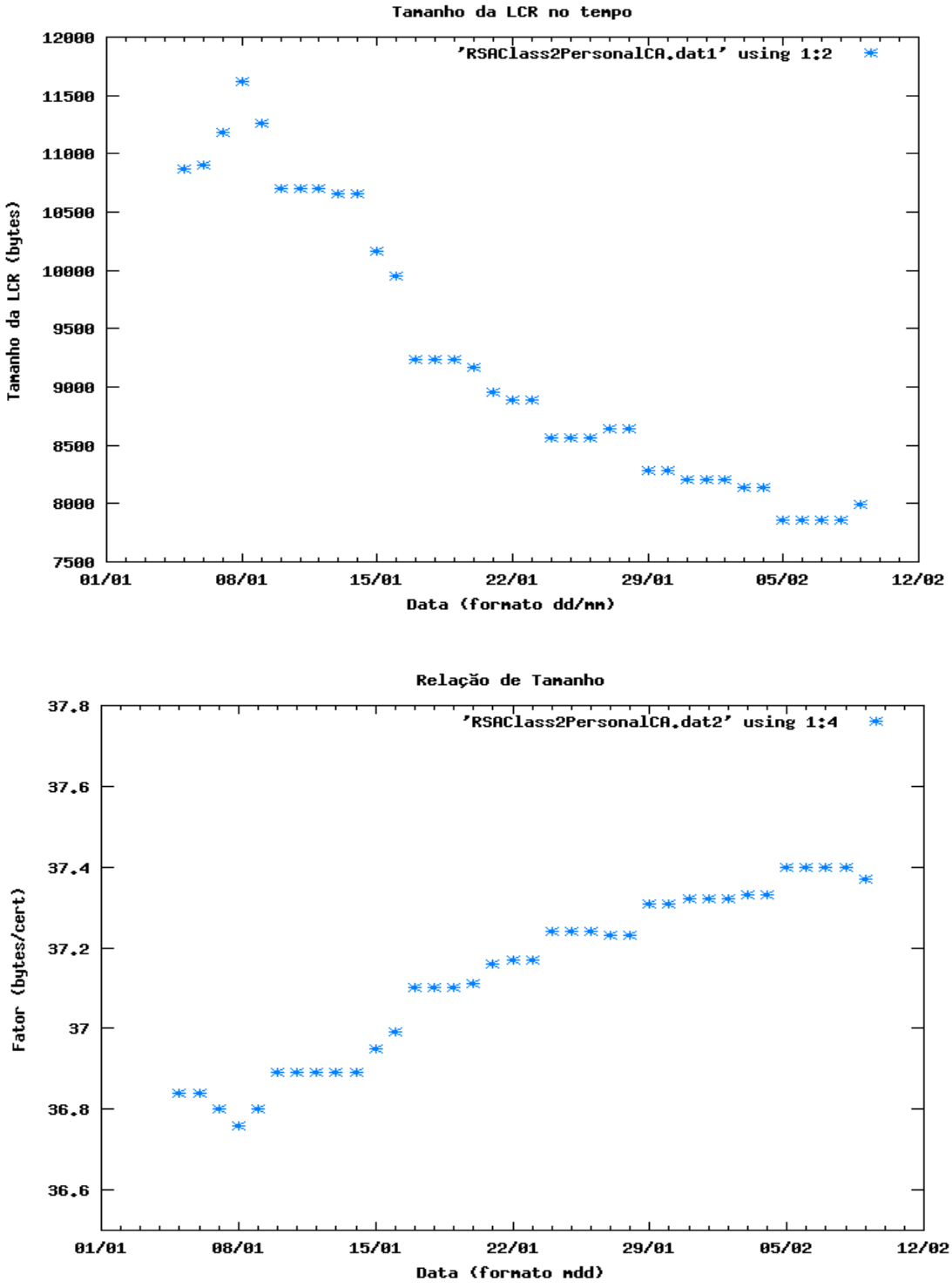


Figura 5-6. Dados da maior LCR da RSA Security

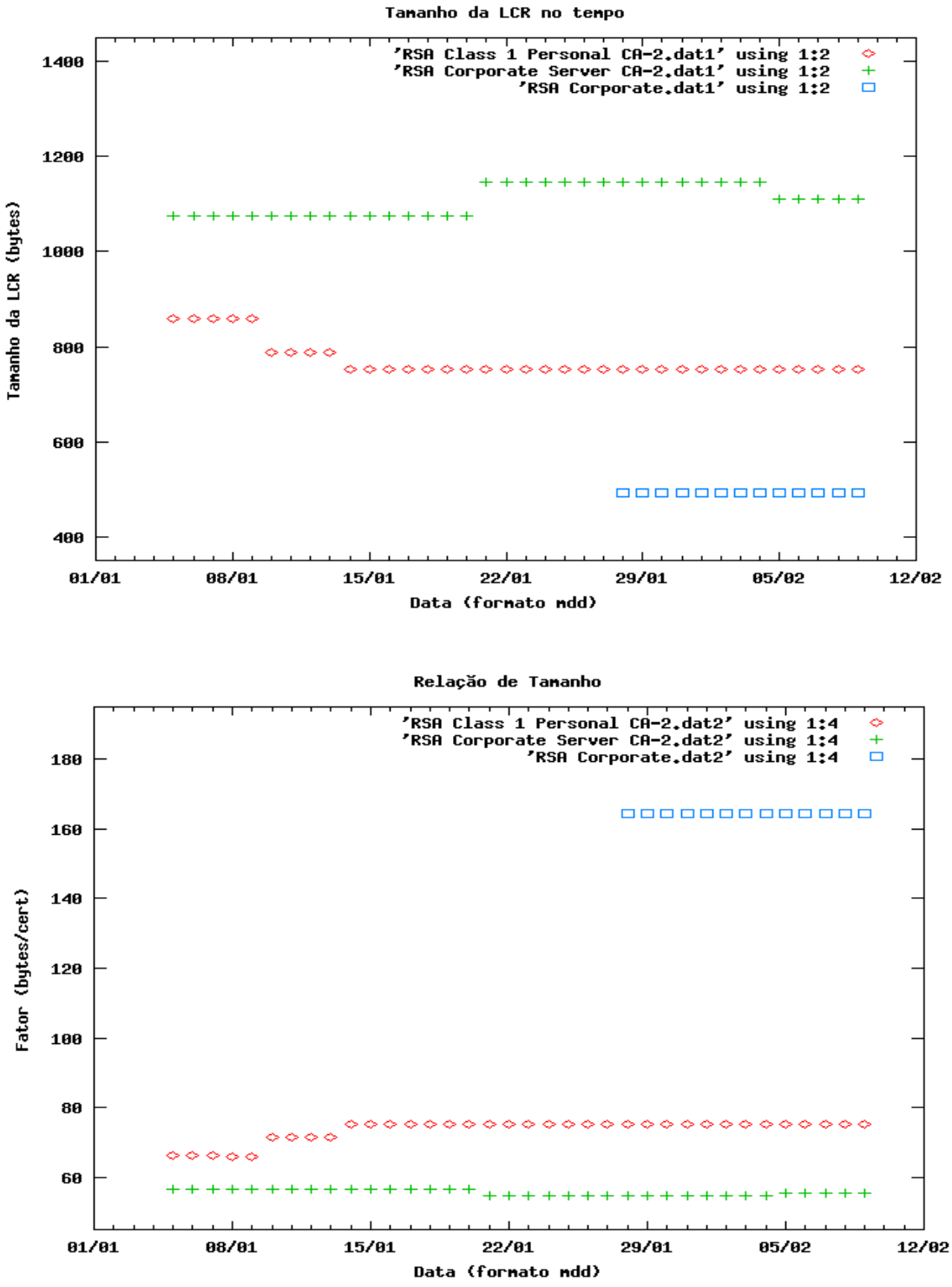


Figura 5-7. Dados das outras LCRs da RSA Security

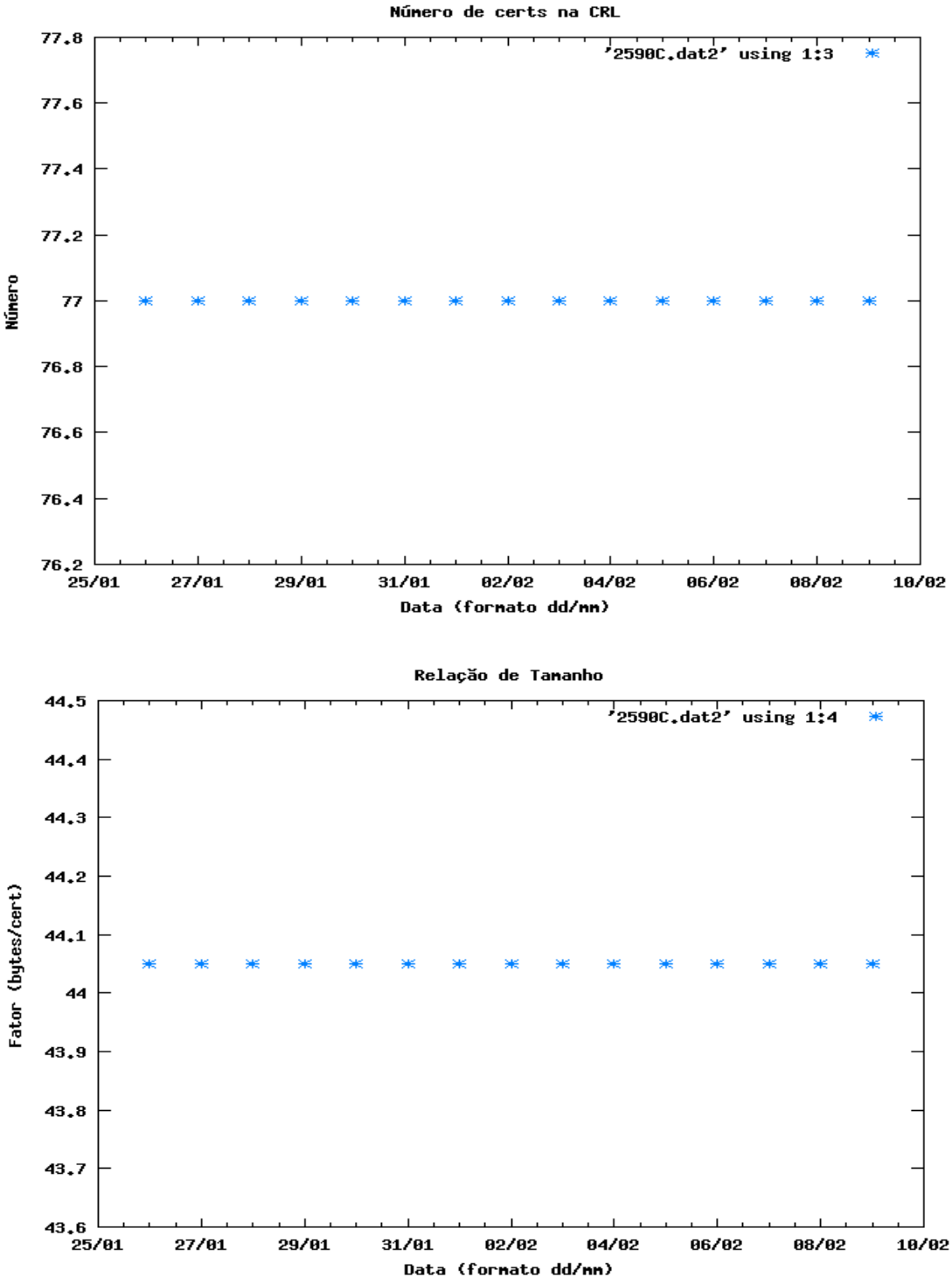


Figura 5-8. Dados da única LCR da Unicert do Brasil

5.1.2 Simulação do Tamanho de uma LCR

As LCRs obtidas das ACs comerciais não fornecem diretamente um parâmetro importante na caracterização dessas listas, a taxa de crescimento da lista em função do número de identificadores de certificados revogados, ou seja, quantos bytes cada identificador de certificado revogado adiciona à lista.

Para obter esta informação foi simulada a evolução de uma LCR contendo certificados com identificadores de número de série com 6 dígitos e assinadas usando uma chave de 1024 bits. Este tamanho foi selecionado pois é o utilizado pela Verisign nos certificados de site seguro emitidos por ela, como pode ser conferido verificando o certificado do site do Banco do Brasil e porque esta classe de certificados revogados é que gera a maior LCR da Verisign, a class3International.crl.

Esse processo foi realizado através da seguinte seqüência:

- utilizando o aplicativo openssl da biblioteca OpenSSL foram gerados por processo automático 2560 certificados, quantidade esta suficiente para obter os resultados desejados (aproximadamente 10 vezes maior que o valor aproximado do ponto de estabilidade da relação entre o tamanho da lista versus a quantidade de certificados nela). Esse procedimento foi desenvolvido para a simulação do esforço computacional para a criação das LCRs, descrito na seção 5.2.1 e os certificados gerados foram aproveitados nesta simulação.

- utilizando novamente o mesmo aplicativo, também de uma forma automatizada e iniciando com 10 certificados e dobrando a cada rodada, foram sendo revogados os certificados e armazenado o tamanho da lista. Com estes valores foram calculadas a relação entre o tamanho em bytes da lista e a quantidade de identificadores de certificados presentes nela para cada rodada.

Devido ao teor determinístico da simulação, não foi necessário aplicar um processo de validação dos resultados. Os dois gráficos que seguem apresentam estes resultados.

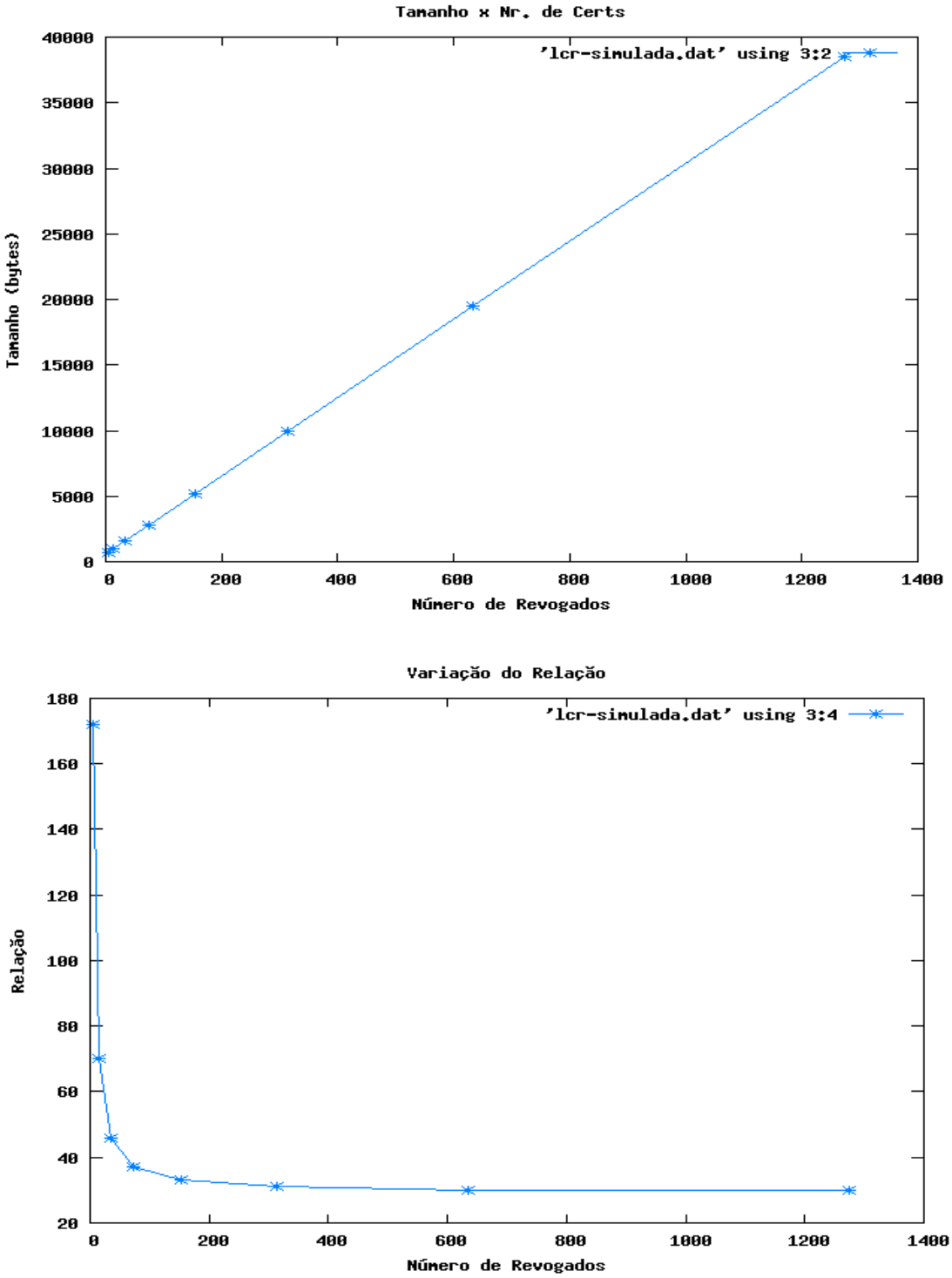


Figura 5-9. Resultado da simulação do tamanho de uma LCR

5.1.3 Tamanho das Mensagens OCSP

A caracterização do tamanho das mensagens trocadas pelo protocolo OCSP foi dividida em duas partes: uma dedicada ao cliente, responsável pela geração da requisição e outra do servidor, responsável pelas respostas.

5.1.3.1 Tamanho das Requisições

Para obter os valores desejados foram gerados inicialmente três conjuntos de 10 certificados, cada um utilizando uma quantidade determinada de dígitos no identificador único dos certificados, além de um certificado extra que foi sempre o mesmo utilizado em todas as requisições assinadas. Todos os certificados gerados utilizaram chaves de 1024 bits pelo mesmo motivo apresentado na seção 5.1.2.

O processo de geração dos conjuntos foi o seguinte:

- utilizando o aplicativo openssl em um processo em lote com um procedimento automático que alterava apenas a quantidade de dígitos dos identificadores dos certificados, mantendo todos os outros valores inalterados, foram gerados os 3 conjuntos mencionados.

- após este passo, utilizando o cliente desenvolvido com o auxílio da biblioteca cryptlib, foram geradas requisições OCSP, tanto as assinadas quanto as não assinadas, que foram salvas em disco.

Esse último passo foi repetido para cada conjunto, utilizando como tamanho para os identificadores únicos dos certificados os 3 valores encontrados na LCRs obtidas das certificadoras pesquisadas .

Analisando os resultados obtidos verificou-se que não houve variação no tamanho das requisições dentro dos conjuntos gerados. Além disso, verificou-se que houve variação entre os conjuntos e que eram dependentes da quantidade de dígitos do identificador único dos certificados.

A tabela 5-1 resume os resultados obtidos.

Dígitos do número de série	Tamanho das Requisição	
	Assinadas	Não Assinadas
2	1769	103
6	1771	105
32	1784	118

Tabela 5-1. Tamanho da requisição OCSP vs. tamanho do identificador

5.1.3.2 Tamanho das Respostas

Para obter os resultados desta simulação foram utilizados os mesmos conjuntos da simulação da requisição. O certificado extra empregado anteriormente para assinar as requisições foi utilizado com certificado de AC para assinar as respostas e para isso o arquivo contendo a sua chave privada foi necessário.

O procedimento da simulação foi o seguinte:

• as requisições geradas na simulação anterior eram, uma de cada vez, lidas pelo OCSP *Responder* criado utilizando a biblioteca cryptlib.

• para cada uma das requisições que eram processadas o servidor salvava em disco uma resposta assinada contendo uma situação de certificado válido e outra de revogado.

O servidor foi configurado para emitir apenas respostas básicas, ou seja, sem a inclusão de extensões e sem incluir o seu próprio certificado na resposta, o que aumentaria ainda mais o tamanho desta.

Analisando os resultados obtidos verificou-se novamente que não ocorreu variação dentro dos conjuntos e que as variações entre conjuntos dependiam do número de dígitos do identificador dos certificados. Além disso, foi possível verificar que as respostas geradas para certificados com situação revogada ou não tinham tamanhos idênticos.

A tabela que segue resume os resultados

.

Dígitos do número de série	Tamanho das Respostas	
	Assinadas	Não Assinadas
2	1405	-
6	1407	-
32	1420	-

Tabela 5-2. Tamanho em bytes da Resposta OCSP

5.2 Esforço Computacional

Na ausência de um método prático e de fácil utilização para avaliar o esforço computacional tomou-se como parâmetro o tempo de execução dos diversos processos. Foram simuladas as seguintes grandezas:

5.2.1 Tempo de Geração de LCRs

Para obter os resultados desta simulação foram utilizados 2560 certificados contendo identificadores únicos de série dos certificados com 6 dígitos e chaves de 1024 bits.

O processo de geração destes certificados o segue:

• utilizando o aplicativo openssl em um processo em lote foram sendo gerados os certificados, iniciando com 10 unidades e dobrando a cada rodada.

• cada rodada era executada 100 vezes e os tempos obtidos armazenados em arquivo para um processamento posterior.

• após todas as rodadas, um script em PHP desenvolvido para processar o arquivo contendo os tempos de processamento obtidos era executado e calculava a média e o desvio dos resultados obtidos.

O gráfico que segue apresenta os valores médios obtidos.

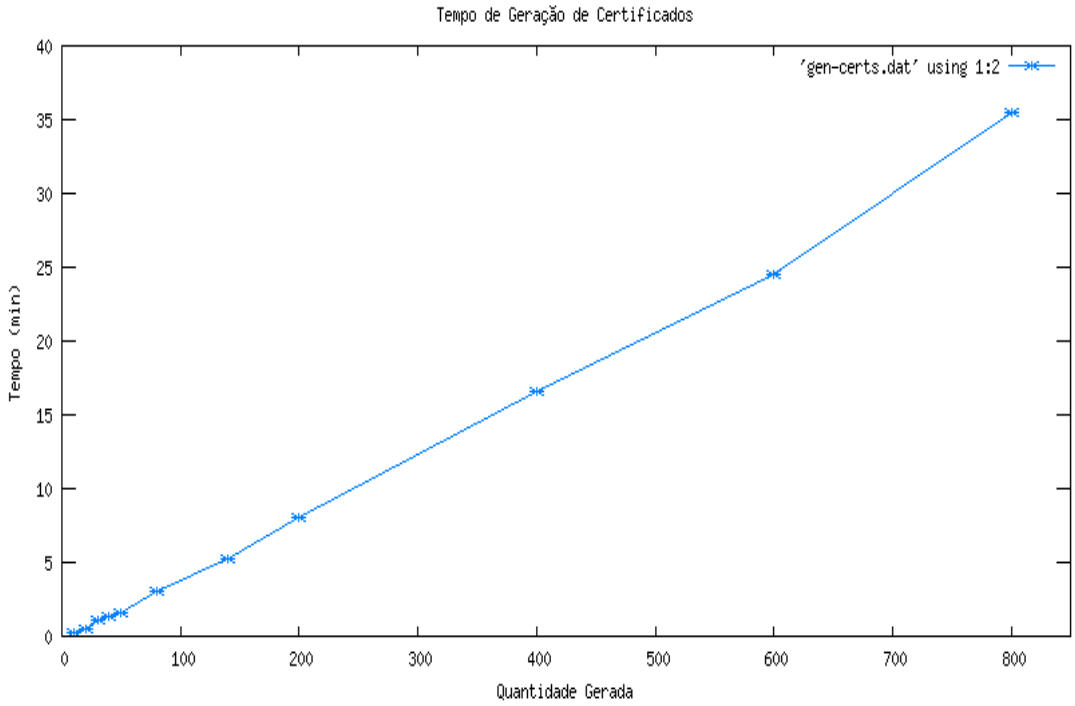


Figura 5-10. Tempo de geração de certificados vs. quantidade

Após a simulação do tempo de geração de certificados em função da sua quantidade foi simulado o tempo de revogação dos certificados também em função da quantidade, só que agora a quantidade se refere ao número de certificados revogados.

O procedimento desta simulação foi o seguinte:

- utilizando o aplicativo openssl num procedimento em lote foram sendo revogados os certificados, iniciando com 10 unidades e dobrando a cada rodada.

- entre cada rodada era executado 100 vezes o processo de geração da LCR. O tempo de cada geração era armazenado em arquivo.

- após todas as rodadas, um script em PHP desenvolvido para processar o arquivo contendo os tempos de processamento obtidos era executado e calculava a média e o desvio dos resultados obtidos.

Analisando os resultados obtidos verificou-se que a variação nos tempos de geração das listas era sempre muito pequena, inferior a 5%.

Os valores médios calculados são apresentados no gráfico que segue.

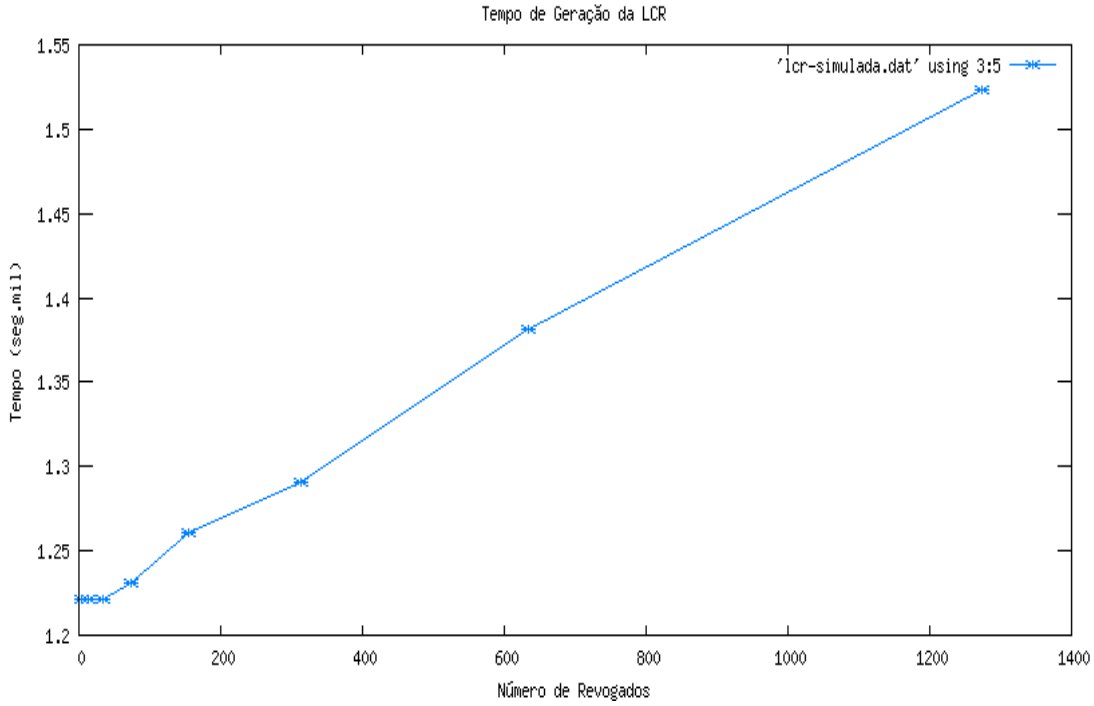


Figura 5-11. Tempo de geração de LCRs vs. número de certificados

5.2.2 Geração das Requisições OCSP

Para obter os resultados desta simulação foi selecionado um certificado de cada conjunto de certificados utilizados na simulação da seção 5.1.3.1.

O processo desta simulação foi o seguinte:

- utilizando o cliente desenvolvido com a biblioteca cryptlib em um processamento em lote, foram fornecidos como parâmetro os certificados, um de cada vez.

- para cada certificado foram criadas 100 requisições e armazenados os tempos de processamento de cada uma delas, para posterior análise.

Analisando os resultados verificou-se que o tempo para a geração de cada requisição praticamente independe da quantidade de dígitos do identificador dos certificados. Além disso, verificou-se também que os valores obtidos nos tempos alternavam-se entre

dois níveis provavelmente devido ao arredondamento dos tempos obtidos. Os resultados são resumidos na figura que segue.

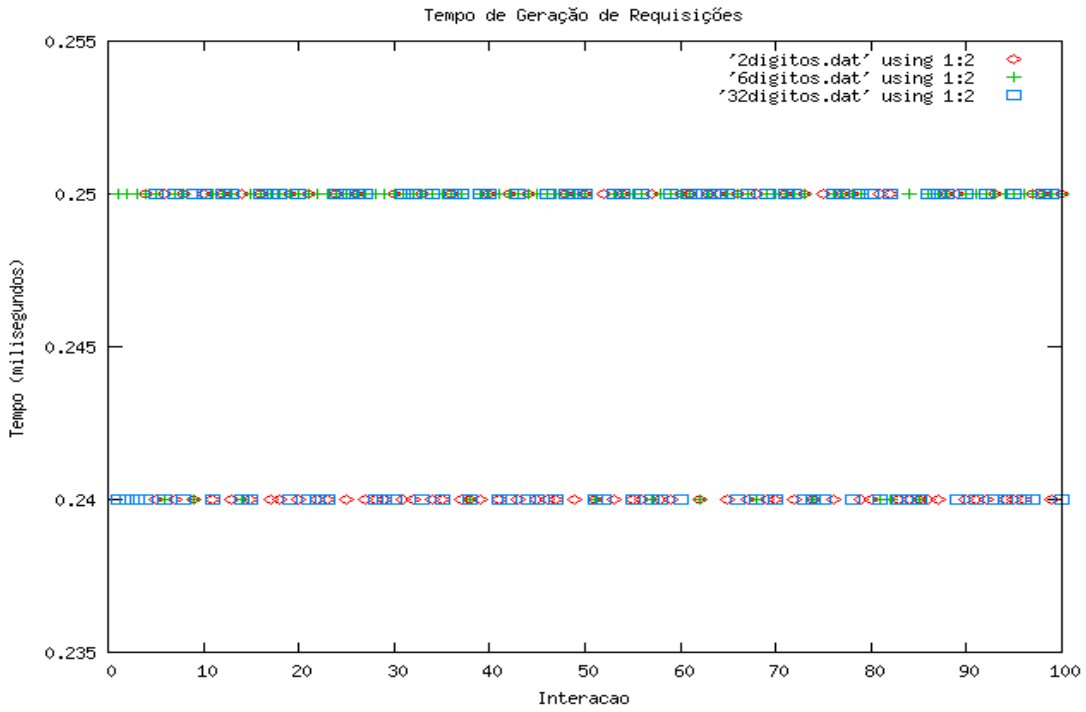


Figura 5-12. Tempo de geração das requisições OCSP

5.3 Análise dos Resultados

A análise dos resultados foi organizada seguindo a divisão em dois grupos feita anteriormente na análise.

5.3.1 Largura de Banda

Para proceder as análises, foram gerados 3 gráficos, parametrizados no tempo, para cada lista das três certificadoras pesquisadas. Os gráficos mostram a evolução do tamanho das listas, a quantidade de certificados nelas e a relação entre estas duas grandezas. Alguns exemplos representativos desses gráficos são apresentados abaixo.

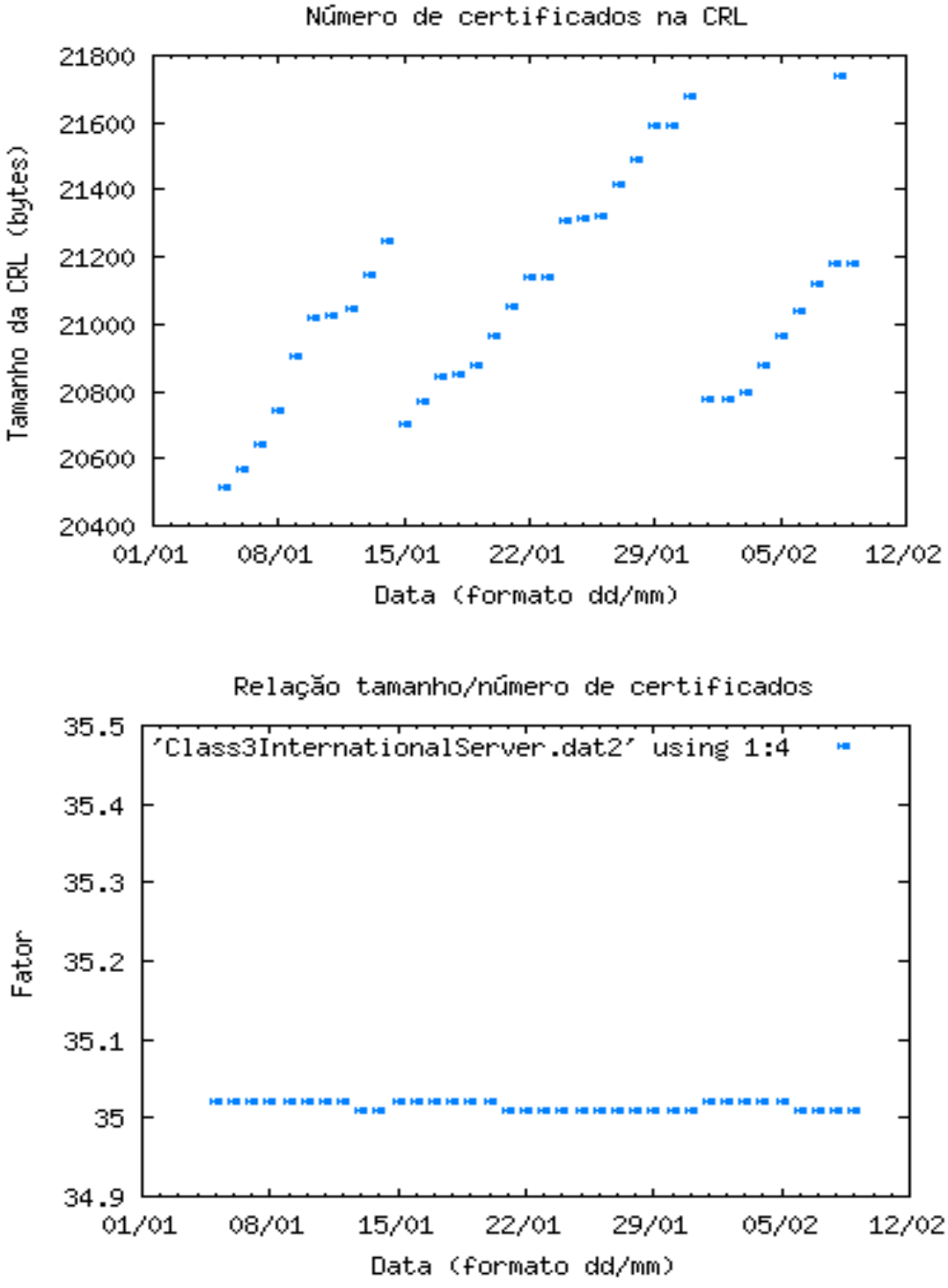


Figura 5-13. Parametros da LCR class3International.crl

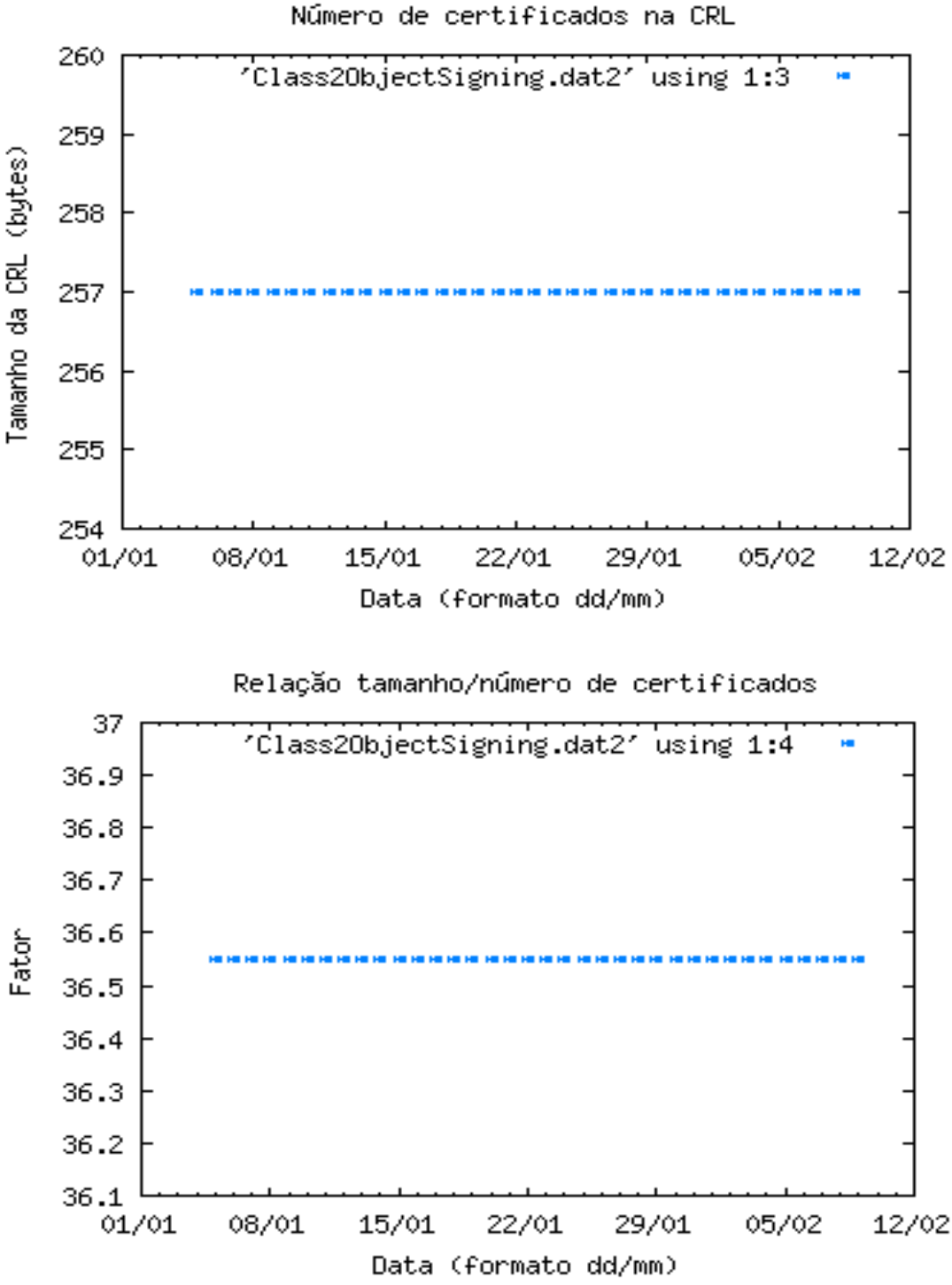


Figura 5-14. Parametros da LCR Class2ObjectSigning.crl

De posse dessas representações e analisando outras informações presentes nos certificados, como a data de promulgação e a de expiração (com as quais pode-se calcular o

perído de validade da lista) foi possível tecer as seguintes conclusões sobre dois pontos cruciais:

5.3.1.1 Atualização das Informações

Existem autoridades que devido ao grande número de certificados gerenciados ou à sua política de certificação executam emissões de listas em intervalos bem menores que a validade das mesmas. Isto é o que ocorre com a maior lista da Verisign, a class3International.crl apresentada na figura 5-13. Essa lista é atualizada no máximo diariamente e possui validade de 14 dias.

Como conseqüencia disso, ocorre que:

- Clientes que possuam a capacidade de ajustar o tempo de consulta à lista, ou seja, consigam ignorar as informação de expiração presente no certificado, terão acesso às informações atualizadas na freqüência que desejarem, até o limite da geração das listas pela autoridade, que pode ser tão curto quanto se deseje.

- Todos os clientes que não possuam essa capacidade (por exemplo, todos os produtos da Microsoft) farão as solicitações na mesma freqüência, determinada pelo intervalo de validade da lista. Como o momento da solicitação da primeira lista para um grande número de clientes independentes pode ser considerado igualmente distribuído, o resultado é que não irá ocorrer picos de demanda pela lista, fazendo com que o consumo de banda devido a solicitações igualmente distribuidas passe a ser não mais dependente do momento das solicitações, mas apenas do tamanho da lista e do número de clientes.

5.3.1.2 Tamanho das LCRs

Por serem essas listas emitidas em intervalos pequenos, torna-se fácil mantê-las enxutas, ou seja, os certificados que tenham sido revogados são removidos da lista pouco tempo após sua expiração. De forma semelhante, os recém revogados são adicionados à lista em média na metade do tempo estipulado para as emissões sucessivas.

Segundo os gráficos gerados, algumas autoridades não consideram relevantes as características da seção anterior e ajustam seus procedimentos para outra realidade. Por

exemplo, a lista RSASecurityServer.crl (vide figura 5-5) parece incluir os recém revogados diariamente, mas remove os expirados apenas na data da expiração da lista.

Com estas informações e as simulações realizadas foi possível caracterizar os fatores que controlam o tamanho das LCRs. As comparações entre os dois métodos serão apresentadas na conclusão ao final deste capítulo.

O primeiro resultado é que o número de bytes acrescidos a uma LCRs estabiliza após aproximadamente 300 certificados na lista, como mostra Figura 5-9, em valores que dependem basicamente dos seguintes parâmetros:

• da presença de extensões nos certificados: para que estas possam ser usadas a versão da lista precisa ser a segunda. Porém somente a mudança da versão e a inclusão de extensões à lista (não em cada entrada revogada) não causam impacto na relação tamanho x quantidade, como foi verificado comparando a LCR RSAClass2PersonalCA.crl da RSA Security que varia entre 315 e 230 certificados (todas as LCRS da RSA Security são emitidas na versão 2 com duas extensões: uma definida na RFC 3280 chamada *Authority Key Indentifier* (AKI) e outra de uso particular identificada apenas pelo seu OID que é 2.5.29.20) com as LCRs Class2ObjectSigning.crl e SecureServerTestingCA.crl da Verisign que possuem número de certificados semelhantes.
A única certificadora a usar extensões por certificado é a Unicert do Brasil (vide Figura 5-8). Essas extensões informam a razão da revogação de cada certificado. A relação nessa LCR (a única emitida pela autoridade) é de 44 bytes/certificado, contendo cerca 77 certificados e usando números de série de 8 dígitos. O fato da LCR conter apenas 77 certificados dificulta a comparação pois não se pode considerar essa lista como tendo atingido uma relação estável.

• o formato do número de série dos certificados: o padrão PKIX impõe apenas que este número não seja repetido, mas nenhuma recomendação é feita sobre o formato deste número. Autoridades que tenham a capacidade de prever o seu espaço de certificados podem reduzir o tamanho das LCRs usando números de série com poucos digitos, como é o caso da Thawte que em seus certificados (ex. ThawteCodeSigningCA.crl - +14 mil certificados) utiliza 6 dígitos para o número

de série e atinge uma relação de 22 bytes/certificado enquanto os certificados da mesma ordem de grandeza da Verisign (ex. Class3InternationalServer.crl - +21 mil certificados) atingem 35 bytes/certificado com 32 dígitos no número de série.
É importante ressaltar que os valores apresentados anteriormente podem ser considerados estáveis devido ao tamanho das listas. Listas menores possuem uma relação bem maior, porém limitadas ao tamanho das listas vazias que podem ter 365 bytes (RSA Business Partner CA.crl - versão 2) ou 412 bytes (ThawtePersonalFreemailCA.crl - versão 1).

Um terceiro parâmetro que verificou-se influenciar no tamanho das LCRs é o conteúdo do campo que armazena a data da revogação. Este parâmetro foi detectado quando foi comparada a relação de bytes por certificado da LCR simulada com uma de tamanho semelhante (ThawteCodeSigningCA.crl). Observou-se que a LCR simulada atingiu uma relação de 30 bytes/certificado enquanto a da Thawte informava 22 bytes/certificado.

Analisando as duas listas percebeu-se que essa diferença é provavelmente devida ao fato de todos os certificados simulados terem sido revogados em uma mesma data, ou seja, foi perdida a variação devida aos tamanhos das datas de revogação, que é composta pelo nome do dia da semana, nome do mês, dia e hora.

5.3.1.3 Mensagens OCSP

Segundo o protocolo e analisando os resultados da seção 5.1.3.1, as requisições feitas pelos clientes variam de tamanho com dois parâmetros principais:

- a presença ou não da assinatura do cliente: as requisições assinadas possuem 1666 bytes a mais que as não assinadas (aproximadamente 17x mais) devido à inclusão da estrutura necessária à mensagem para abrigar os parâmetros e a própria assinatura. Além disso, as requisições, assinadas ou não, aumentam de tamanho a mesma taxa: 1 byte a cada 2 dígitos no número de série.

- o formato do número de série: a variação na quantidade de dígitos no número de série do certificado também foi aplicada as mensagens OCSP que se mostraram pouco variantes aos mesmos parâmetros considerados nas LCRs.

Analisando os resultados verificou-se que:

• As respostas variam da mesma forma que as requisições e possuem tamanho equivalente às resquisições assinadas.

Segundo a definição do protocolo, todas as respostas válidas precisam ser assinadas, o que justifica o aumento considerável no tamanho da mensagem.

5.3.2 Esforço Computacional

5.3.2.1 Tempo de Geração de LCRs

O gráfico da Figura 5-9 mostra que o tempo de geração das LCRs varia quase linearmente com o número de certificados a serem incluídos na lista. Além disso, nota-se que esta geração é extremamente rápida.

5.3.2.2 Geração das Requisições OCSP

O gráfico da Figura 5-12 mostra que o tempo de geração das requisições, independentemente da quantidade de dígitos do número de série, alterna entre dois únicos pontos tendo como média 0.245 milisegundos.

5.3.2.3 Geração das Respostas OCSP

O tempo de geração das respostas OCSP não foi calculado pois é muito semelhante ao tempo de geração das requisições assinadas. Analisando o processo é possível perceber que após receber a requisição, o servidor executa uma procura em algum repositório de informações dos certificados usando como chave o número de série do certificado presente na requisição e após localizá-lo, formata e assina a resposta do mesmo modo executado na geração da requisição. Em seguida a requisição é enviada ao cliente.

5.3.3 Conclusões

A geração de uma LCR sem nenhum critério de controle por certo resultaria em uma lista desnecessariamente grande, contudo a geração da forma adequada consegue tratar o problema eficientemente, e mesmo em grandes ambientes, o uso das LCRs torna-se viável. Este fato aliado ao pequeno esforço de processamento na sua geração e processamento faz dela uma boa opção para a divulgação de informações de revogação. Porém,

alguns parâmetros podem ser ajustados para garantir que seu tamanho permaneça limitado a um valor aceitável, por exemplo:

• a quantidade de dígitos do número de série pode ser reduzida com base no universo de usuários. Em certas aplicações, por exemplo quando o certificado é usado como chave de acesso a um serviço com duração pre-estabelecida, pode-se até reutilizar os números, feitos os devidos controles.

• Reduzir o tamanho da chave utilizada para assinar os certificados. Em geral chaves menores fazem com que os processos ocorram mais rapidamente (vide Figura 5-15) e além disso por possuírem representação menor, ocupam menos bytes nas LCRs geradas. Esta medida deve ser adotada com os devidos controles pois quanto menor o tamanho maior é o risco associado à sua utilização.

• Reduzir o tempo de validade dos certificados: este parâmetro contribui para que os certificados após revogados saiam rapidamente da lista, contribuindo para a sua diminuição.

• Gerar as LCRs sem os campos de extensão: isso por certo reduz seu tamanho e tempo de processamento por aliviar a estrutura a ser processada.

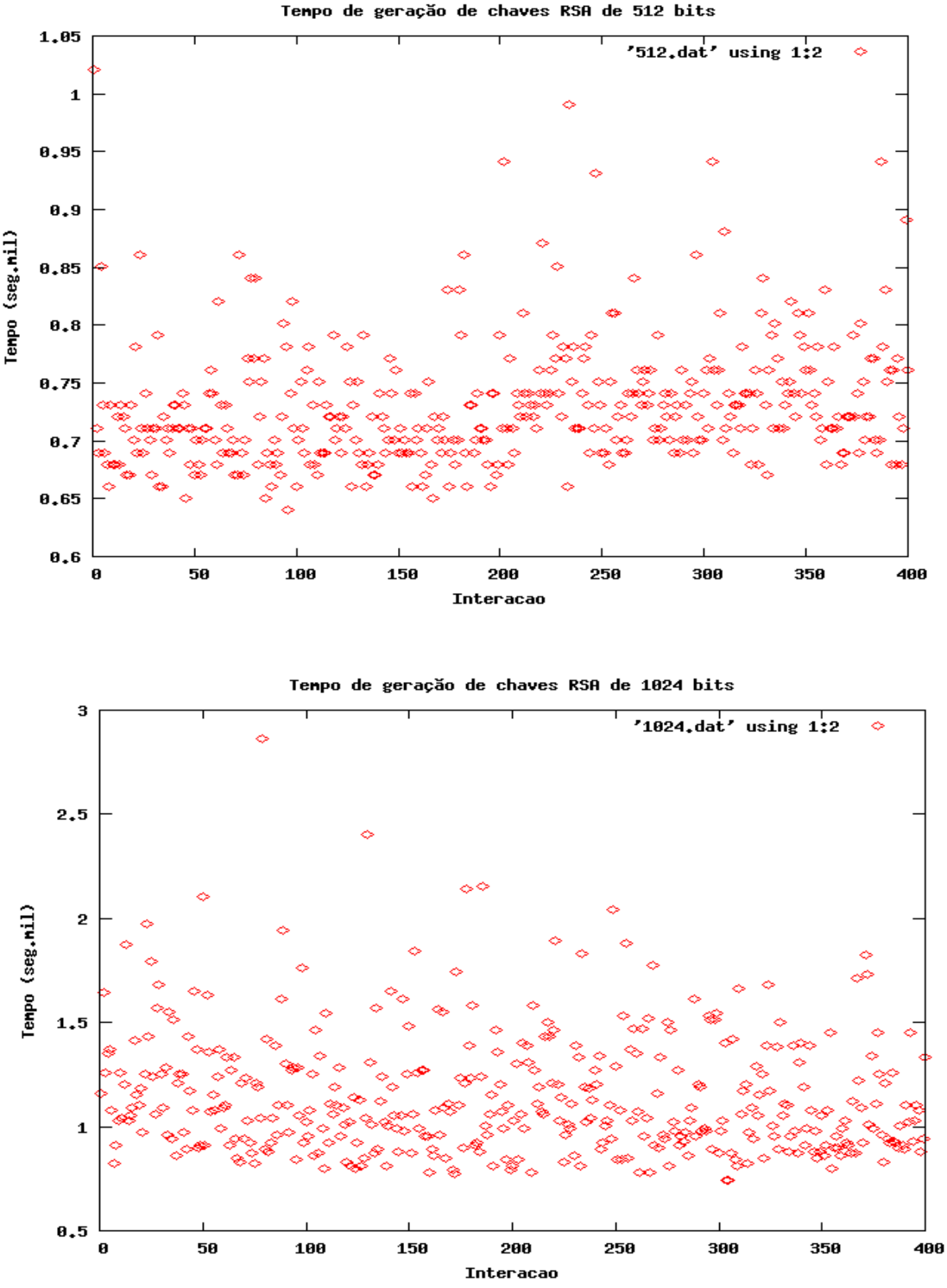


Figura 5-15. Tempo de geração de chaves RSA de 512 e 1024 bits

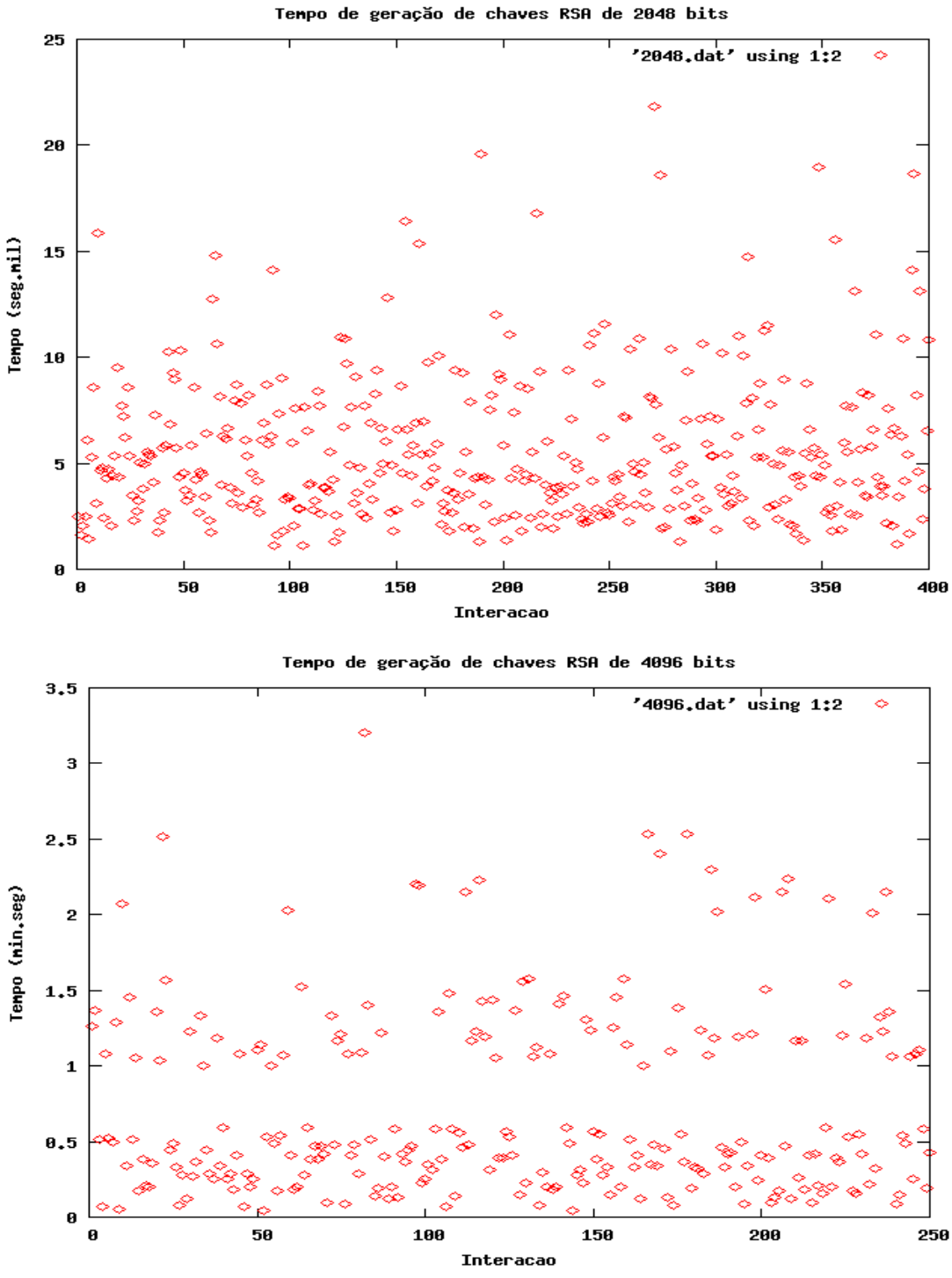


Figura 5-16. Tempo de geração de chaves RSA de 2048 e 4096 bits

Alguns controles semelhantes podem ser feitos com as requisições das LCRs de modo a controlar o consumo de recursos da rede, por exemplo:

• Redirecionamento de servidor, que é um processo simples e eficiente quando utilizando o protocolo HTTP para a obtenção das LCRs, gerando uma forma de balanceamento de carga e tráfego.

• controlar o tempo de promulgação das listas baseando-se na demanda. Listas com poucos certificados podem ser promulgadas mais freqüentemente, fazendo com que os cliente tenham sempre informações atualizadas a um baixo custo de processamento e de transporte. Com o aumento do tamanho das listas, pode-se reduzir o tempo das promulgações e em paralelo a isso, utilizar um protocolo de verificação online como o OCSP. Dependendo da implementação do cliente, os dois processos podem ser conjugados.

Uma desvantagem percebida das LCRs está na baixa funcionalidade fornecida. Prevendo esta limitação, a segunda versão padronizada, que não foi analisada pois ainda não entrou em amplo uso, adiciona alguns mecanismos que corrigem em grande parte o problema, utilizando a mesma solução que evoluiu o padrão dos certificados da versão 1 para a 2, usando extensões.

Um outro problema do uso das LCRs é um maior grau de dificuldade na criação de aplicativos. Para usá-las, torna-se necessário localizar o seu servidor, baixar a lista e processá-la. Durante algum momento neste processo, é necessário verificar se ela ainda não expirou e neste caso obter e armazenar uma nova. A operação com os protocolos online, tipo o OCSP analisado, é bem mais simples, a cada necessidade de validação, cria-se uma requisição que é enviada a um servidor e ao receber a resposta toma-se a decisão sobre a validade. Não existe a necessidade de armazenar nenhum material e nem de verificar o final da validade da lista, em contra-partida o processos só pode ser executado online.

Concluindo, ainda que não tenha sido possível avaliar fielmente o impacto na rede das duas opções estudas, pois para isso seria necessário saber as características da distribuição das requisições ao longo do tempo, pode-se comparar o tamanho das LCRs com a soma das requisições com as respostas do OCSP.

A tabela que segue mostra a quantidade de certificados revogados que poderiam ser transportados em LCRs em diferentes estágios consumindo a mesma quantidade de bytes

das mensagens de requisição e resposta do OCSP nos dois casos possíveis: com requisições assinadas e não assinadas.

Mensagens OCSP[a]	Tamanho em bytes	Relação Tamanho/Número de Certificados na LCR[b]						
		93.6(1)	62(2)	53.5(3)	37,9(4)	36.5(5)	35,3(6)	22(7)
Req + Resp	1538	16.5	24.8	28.8	40.5	42.1	43.5	69.9
Req_Sig + Resp	3204	34.2	51.7	59.9	82.5	87.8	90.8	145.6

Tabela 5-3. Relação tamanho/número de certificados em uma LCR

a. Tomando por base números de série com 32 dígitos

b. Listas de Referencia: (1)BTClass1Individual.crl - 5 certs. (2)Class3IntranetServer.crl - 15 certs (3)Class3WLANServer.crl - 21 certs(4)Class2ObjectSigning.crl - 128 certs (5)Class3NewOFX.crl - 277 certs (6)Class3CodeSigningCA2001.crl - 1140 certs (7) ThawtePersonalFreemailIssuingCA.gif - 4200 certs

Num cenário onde os certificados digitais fossem usados apenas para a autenticação e acesso a recursos, como é o caso do ambiente do AirStrike, ocorreria por tentativa de acesso apenas uma rodada de checagem. Se para isso fosse utilizado o protocolo OCSP com requisições assinadas (sem a qual não haveria autenticação mútua de forma eficiente) seriam necessários aproximadamente 3200 bytes. Com este valor seria possível:

- obter toda uma lista com aproximadamente 6 certificados,
- obter aproximadente 12 certificados de uma lista com 15, ou
- obter 14 certificados de uma lista de 21, ou
- obter 20 certificados de uma lista de 277.

Assumindo que em um determinado instante 10% do número de certificados emitidos estivessem revogados, para o primeiro caso, o número de usuários seria de aproximadante 60, que é um número próximo ao número comum de conexões simultâneas a um ponto de acesso de uma rede sem fio (que é de 64 conexões). Neste cenário, utilizando as LCRs, todos os usuários e o ponto de acesso poderiam utilizar os certificados de forma seguraentre si (pois saberiam da situação de validade dos mesmo) sem a necessidade de outras solicitações.

Capítulo 6

Ambiente de Teste: AirStrike

6.1 Introdução

A motivação para a criação do sistema AirStrike por Carrion[57] gira em torno das falhas nos atuais mecanismos de segurança das redes sem fio, principalmente no protocolo WEP. Com a insegurança desse protocolo outras soluções de segurança precisam ser implementadas.

O objetivo do AirStrike é garantir a segurança das redes sem fio baseadas no padrão 802.11b através do desenvolvimento de protocolos, ferramentas e metodologias de segurança[8][13].

A implantação de uma rede sem fio interligada a uma rede cabeada requer a utilização de um gateway, chamado de Ponto de Acesso (ou *Access Point* - AP). Neste ambiente híbrido, onde parte das informações trafegam por um meio não confinado, existem novos desafios, como garantir a autenticação, a autorização e o sigilo do tráfego aéreo e dessa forma proteger o outro tráfego, o presente no segmento cabeado, isto sem perder o forte apelo da mobilidade existente nessas novas formas de rede.

O AirStrike considerou a questão do controle de acesso dos usuários. A arquitetura proposta e implementada (vide Figura 6-1) faz uso de firewalls e endereçamento IP não roteável tornando possível conter e controlar o tráfego nas redes sem fio.

Em resumo, o AirStrike é um sistema de controle de acesso de usuários a uma rede cabeada através de uma rede sem fio interligada a esta pelo Ponto de Acesso, provendo uma estrutura de segurança necessária para que este acesso ocorra de forma segura e eficiente para o cliente.

Os componentes principais do AirStike são:

- AirStrikeAP: contém as funcionalidade do ponto de acesso IEEE 802.11b e

os serviços de DHCP, Firewall, NAT, HTTP e o servidor do método de detecção de desligamento de estação (DPD - *Dead Peer Detection*) que é chamado de isAlive-Daemon.

• AirStrikeClient: contém as funcionalidades das estações que irão se associar a uma WLAN AirStrike. Incluso o isAliveClient que é o componente necessário aos clientes para que estes possam informar periodicamente ao isAliveDaemon sobre sua permanência na rede.

• AirStrike PKI: infra-estrutura de chaves públicas utilizada no âmbito do sistema para prover os certificados de identificação e também os métodos de verificação da situação destes certificados.

• AirStrikeDatabase: sistema de gerenciamento de banco de dados no qual serão armazenadas as informações de autenticação do sistema.

O detalhamento do funcionamento da arquitetura proposta será apresentado na seção que segue.

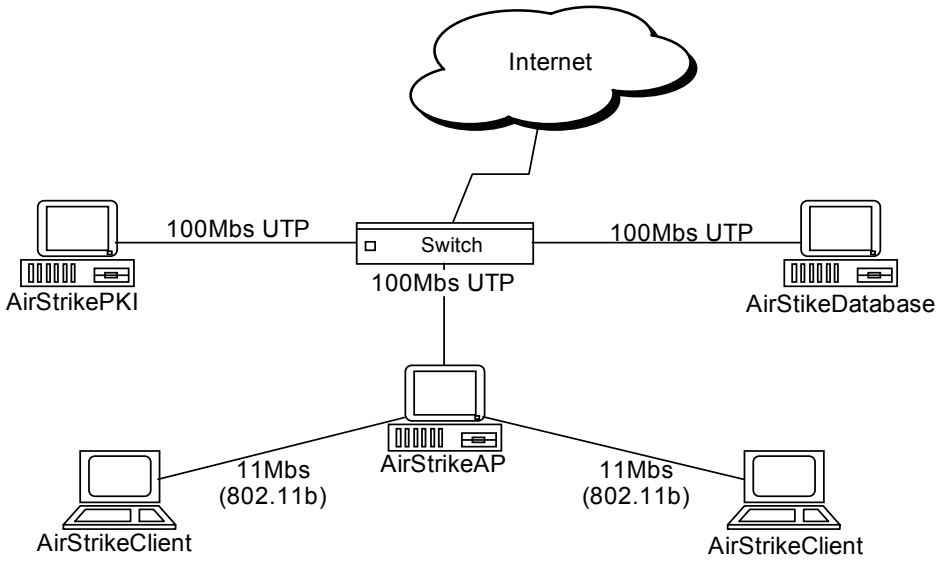


Figura 6-1. Arquitetura original do AirStrike[9]

9. Versão adaptada da figura 1 da referência [8]

6.2 Funcionamento do ambiente AirStrike

Com o objetivo de atingir os controles de segurança desejados, diversas medidas foram adotadas na configuração do AirStrikeAP, algumas relacionadas com a segurança do próprio sistema operacional, e outras relativas aos mecanismos para a autenticação, autorização, sigilo e integridade das informações transmitidas pela rede sem fio.

Para alcançar estes objetivos, uma série de mensagens são trocadas entre o AirStrikeAP (referenciado por simplicidade como AP) e a estação móvel (referenciada com STA). A relação apresentada a seguir foi transcrita na íntegra de [8] para auxiliar na compreensão do processo e na relação do mesmo com os certificados digitais. A Figura 6-2 apresenta de forma gráfica a sequência.

i. STA procura uma rede no domínio definido pela SSID

ii. STA e AP se sincronizam e a associação é estabelecida

iii. STA requisita um IP (cliente DHCP)

iv. AP fornece um IP à STA (servidor DHCP)

v. STA envia mensagens UDP para formação do túnel privado virtual (VPN)

vi. AP verifica credenciais do usuário móvel, formando o túnel VPN

vii. Usuário da STA acessa página web a fim de se autenticar

viii. O AP requisita o certificado digital do usuário da STA

ix. O usuário da STA apresenta o seu certificado digital

x. O AP apresenta o seu certificado digital à STA

xi. O AP acrescenta o IP da STA no banco de dados de IPs autorizados

xii. O AP reconfigura as regras de firewall

xiii. A STA está pronta para utilizar os recursos de rede através do AP

xiv. Durante o período de conexão, o AP verifica se a STA continua ativa, a fim de que possa controlar de forma adequada as regras do firewall.

A utilização dos certificados nos ítens viii a x requer a existência de uma ICP para que estes possam ser gerados e validados. Por uma decisão de projeto, o processo de autenticação do cliente foi divido em cenários, apresentados a seguir.

No primeiro cenário o cliente fornecia para autenticação no AP o par login e senha através de uma interface WEB e recebia do AP uma resposta e o seu certificado que não era verificado. Neste contexto, pelo fato do AP não receber o certificado do usuário e este não verificar o certificado do AP não era possível autenticar mutuamente as partes que participavam do processo de autenticação tornado possível o ataque apresentado na Figura 6-7. Isto ocorreu devido às limitações na implementação inicial do AirStrike-PKI[13].

Outros cenários planejados consideram a troca e validação de certificados, mas para tais cenários serem realizados tornava-se necessário expandir as funcionalidades do AirStrikePKI, o que foi realizado nesta dissertação com uma nova implementação deste componente do sistema, agora utilizando a solução de ICP selecionada anteriormente em conjunto com as implementações do cliente e servidor OCSP.

Com a implantação da autoridade certificadora selecionada e a inclusão da implementação do cliente e servidor OCSP tornou-se possível realizar completamente a arquitetura original proposta. Além disso foi possível expandir os cenários disponíveis para validação dos usuários e alterar a forma de autenticação do DPD, que possuía um ponto frágil, como será visto nas seções seguintes.

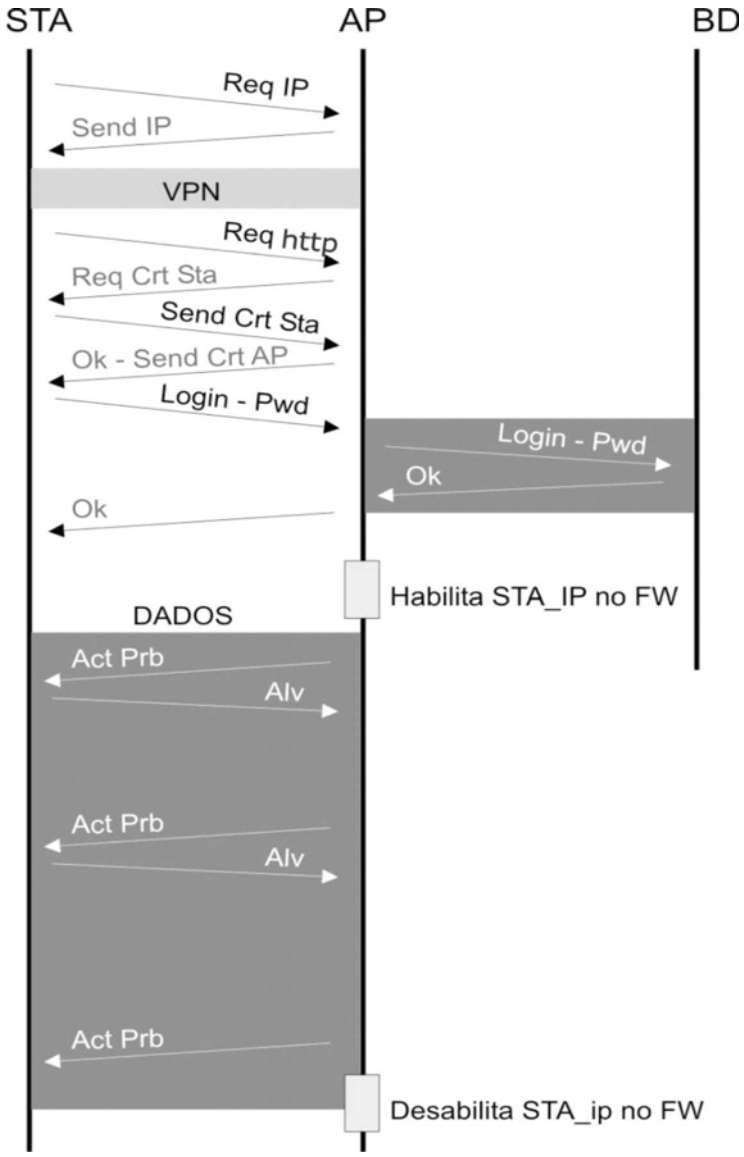


Figura 6-2. Seqüência de mensagens no ambiente AirStrike[10]

6.3 Cenários de Operação

Como mencionado na seção anterior, no primeiro cenário de operação do sistema não exigia o envio do certificado do usuário e este não validava o certificado recebido do AP. A figura que segue representa graficamente o cenário.

10. fonte: figura 2 da referência [8]

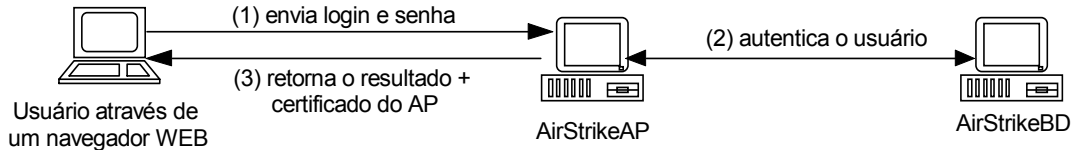


Figura 6-3. Cenário de autenticação do AirStrike sem certificados

Neste cenário qualquer navegador web poderia ser utilizado para a autenticação do usuário. Um pré-cadastro do usuário no sistema de autenticação (composto pelo AirStrikeAP e pelo AirStrikeBD) era necessário. Em geral a manutenção de sistemas de cadastro como este com muitos registros, composto por dados pessoais e de acesso, acarretam algumas dificuldades de gerenciamento e segurança. Além desse problema, existe o ataque apresentado na Figura 6-7 onde um atacante pode, com os devidos acessos, se fazer passar por um usuário legítimo.

Com a entrada em operação da nova implementação da autoridade certificadora AirStrikePKI (utilizando a solução desta dissertação) foi possível criar novos cenários de acesso mais seguros, flexíveis e de gerência mais fácil, alguns desses cenários são apresentados a seguir.

6.3.1 Autenticação usando Certificados e LCRs

Num cenário utilizando certificados digitais o paradigma de autenticação e de gerenciamento se alteram consideravelmente. Por exemplo, na questão do gerenciamento no cenário anterior os dados de identificação dos usuários bem como os utilizados para o acesso são mantidos num banco de dados, que precisa ser protegido, estar sempre consistente e sofrer freqüentes backups. O descredenciamento de acesso a um usuário é uma informação crucial ao processo e neste cenário sem certificados precisa ser armazenado também nessa base de dados e ser consultado periodicamente senão pode ocorrer de um usuário se manter autenticado mesmo tendo sido este privilégio removido.

Com a adoção dos certificados, ainda é necessário um tipo de pré-cadastramento do usuário (na verdade o usuário precisa ter um certificado e para tal deve solicitá-lo em geral através de um formulário, como o apresentado Figura 3-3), porém as informações necessárias para o seu acesso e sua identificação não precisam mais ser mantidas em alguma

base pois todas elas podem ser incluídas no próprio certificado gerado. Com isso o custo de gerenciamento dessas informações tende a ser menor e a segurança do processo maior.

Na emissão do certificado pode-se estipular um tempo de validade para este e desta forma cria-se um mecanismo onde pode-se usar o certificado como um ticket de acesso por tempo determinado.

Se durante esse período de validade ainda for necessário cancelar esse acesso basta revogar esse certificado. Em geral o processo de verificação da situação do certificado é feito numa freqüência pré-estabelecida que pode ser tão pequena quanto se deseje e é feito consultando-se uma LCR emitida pela AC contendo os certificados que foram revogados. A quase totalidade dos navegadores web e outros clientes, como alguns programas de email, já possuem o suporte necessário para executar todo esse processo.

Esta nova forma de validação pode gerar dois novos cenários:

• o primeiro é semelhante ao anterior com a única diferença de que neste o certificado do AP é verificado usando para isso uma consulta a uma LCR emitida pela AC que gerou o certificado do AP. Neste cenário (vide Figura 6-4, a numeração mostra a ordem das mensagens) ainda existe o problema da autenticação do cliente pois usa-se o método de login e senha, o que motiva o próximo cenário.

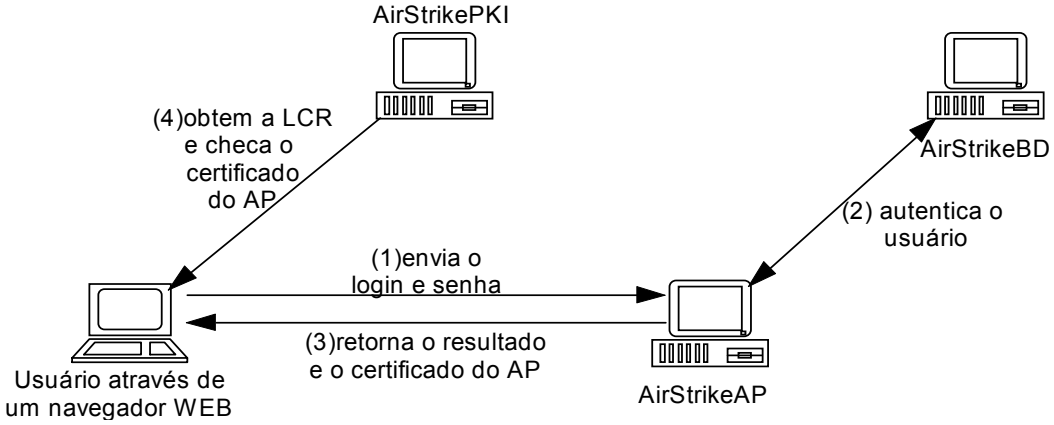


Figura 6-4. Cenário de autenticação exclusíva do AP usando LCR

• Neste segundo cenário pode-se corrigir o problema de autenticação do usuário substituindo o uso do par login e senha pelo envio do certificado do usuário. Gráficamente este cenário pode ser esquematizado da seguinte forma :

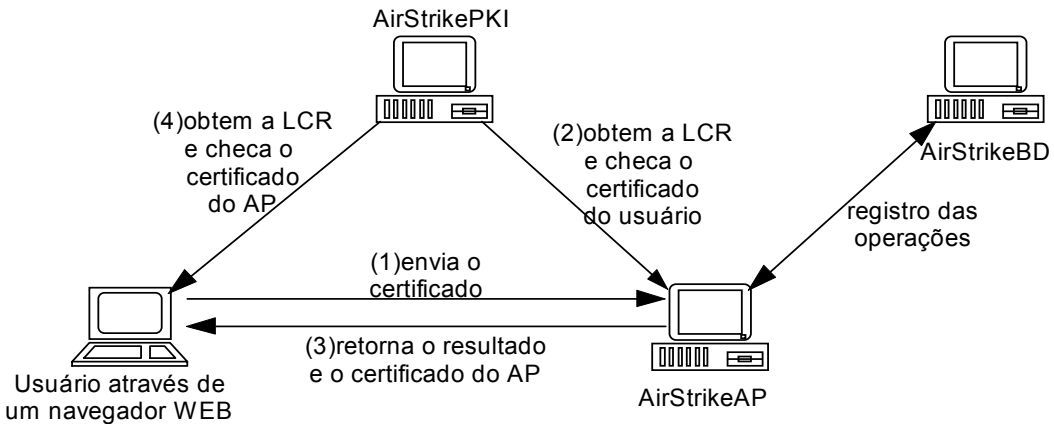


Figura 6-5. Cenário de autenticação mútua usando LCRs

Mesmo com a utilização exclusiva dos certificados, pode-se ainda optar pelo uso do mecanismo de login e senha somente para manter uma interface conhecida pelo usuários.

Nos cenários utilizando exclusivamente os certificados digitais o AirStrikeBD passa assumir a função de *log billing* (registrador de operações executadas).

6.3.2 Autenticação usando Certificados e OCSP

Os cenários apresentados anteriormente também podem ser utilizados com o protocolo OCSP com a vantagem de não mais depender da existência de uma LCR (cujos problemas foram relacionados no capítulo 2). Além disso, em alguns casos, as LCRs podem chegar a um tamanho maior que as mensagens OCSP e desta forma o uso deste protocolo pode reduzir o tráfego sobre a rede.

O cenário anterior, com o uso exclusivo dos certificados, utilizando o OCSP pode ser esquematizado pelo seguinte diagrama.

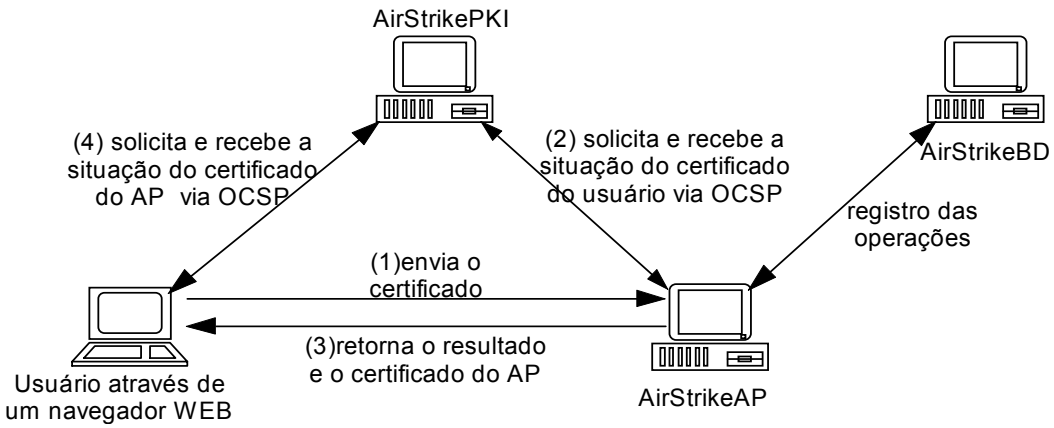


Figura 6-6. Cenário de autenticação mútua usando OCSP

O inconveniente deste protocolo é que ele ainda não é totalmente suportado pelos navegadores WEB. Somente o Netscape 4.7+ ou o Mozila 1.0+ suportam esse protocolo nativamente. Nenhum produto da Microsoft possui suporte nativo para este protocolo.

Como solução para esse problema de suporte, o cliente desenvolvido nesta dissertação pode ser integrado com outras aplicações e utilizado num método *standalone* de autenticação de forma totalmente independente de qualquer outro aplicativo.

Nestes cenários em geral, torna-se possível:

• autenticar o AirStrikeAP e o cliente de forma efetiva, ou seja, uma vez emitidos os certificados para cada uma das partes, é possível comprovar a identidade de cada uma delas mútuamente, evitando assim um ataque clássico de impersonalização (também conhecido com *man-in-the-middle*) representado na Figura 6-7. Na verdade, ao utilizar uma chave pública, uma entidade precisa ter a garantia de que esta chave realmente pertence à entidade com a qual se deseja comunicar. Sem tal certeza, é possível obter uma chave pública de alguém se fazendo passar por outra

entidade.

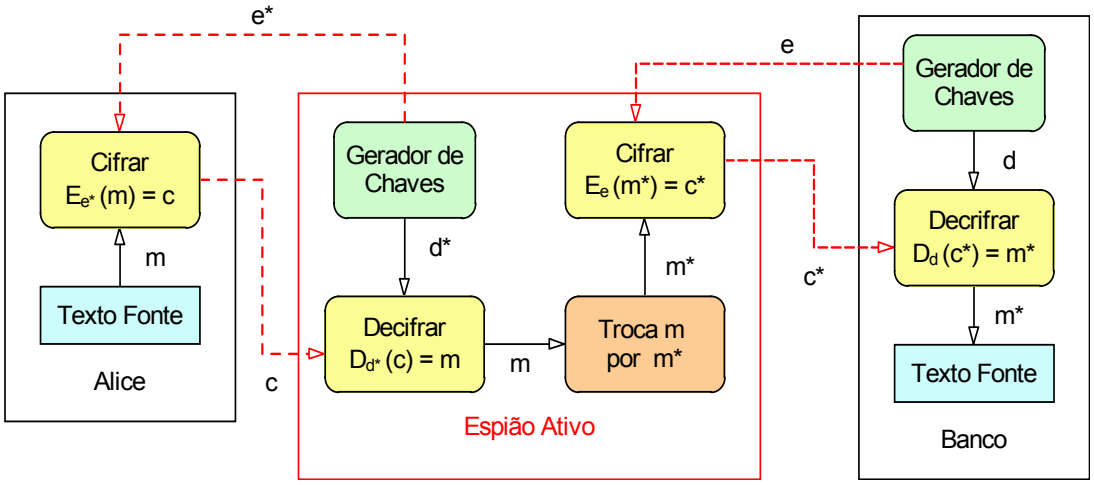


Figura 6-7. Esquema do ataque *man-in-the-middle*.

A autorização utilizando certificados digitais pode ocorrer de várias formas. Uma delas faz uso dos navegadores web. Neste caso configura-se uma página de acesso que quando acessada envia o certificado do servidor, no caso o próprio AP, e solicita o do cliente. Em seguida ambos verificam, cada um por seu próprio método, por exemplo consultando uma LCR ou um servidor OCSP, se os certificados são válidos e confirmada a validade está estabelecida a confiança entre as partes. Outras formas podem alterar apenas a fase inicial do processo, ou seja, a forma como os certificados são trocados. Todo o processo que segue é basicamente o mesmo. Com essa troca do processo inicial, outros clientes podem ser usados, como programas legados com suporte aos certificados ou aplicativos especialmente desenvolvidos para tal fim.

É possível verificar que, mesmo com estas variações, o processo é muito semelhante ao apresentado entre as fases viii a x da descrição da arquitetura, o que demonstra que a utilização dos certificados digitais não altera substancialmente a arquitetura proposta e além disso agrega novas funcionalidades pois elimina a necessidade do cliente acessar uma página web e fornecer um login e senha, que em geral, são difíceis de serem gerenciados em um ambiente com grande número

de usuários.

• corrigir algumas deficiências do método de detecção de desligamento de estação implementado. O DPD originalmente implementado no AirStrike (vide Figura 6-10) utiliza para autenticação mútua uma chave pré-estabelecida (método conhecido como *Pre-Shared Key* - PSK). O processo implementado usando essa PSK possui falhas semelhantes às do processo de comunicação segura empregando criptografia simétrica apresentado no Apêndice I. O fato dessa chave ser a mesma para todos os clientes (e servidores) permite que falsos servidores sejam facilmente criados e do mesmo modo, falsos clientes podem assaltar uma seção e continuar se comunicando com o servidor sem que este identifique esta mudança. Para resolver este problema pode-se utilizar o certificado do cliente em substituição à chave compartilhada e deste modo corrigir as falhas existentes.

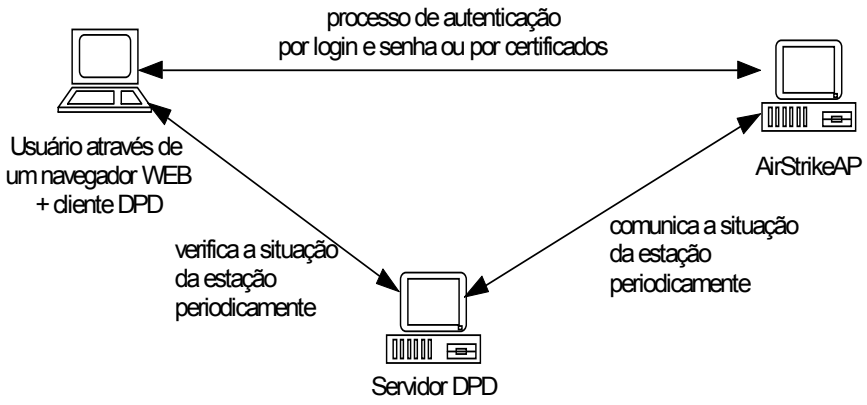


Figura 6-8. Funcionamento do DPD no AirStrike

A comparação da arquitetura do AirStrike com as outras existentes demonstra a grande diferença entre os estágios de desenvolvimento. Em geral, as outra soluções existentes (Oasis[54], NoCat[55] e NetLogon[56]) ainda não chegaram ao ponto de implementar uma proteção para seus métodos de detecção de desligamento de estação. A utilização dos certificados digitais na arquitetura do AirStrike faz com que esta se torne ainda mais segura, se distanciando em qualidade ainda mais das outras soluções.

6.3.3 Realização dos Cenários

Para realizar o primeiro cenário proposto (que pode ser visto como a autenticação do servidor frente aos clientes usando LCR) a Autoridade Certificadora foi configurada de modo que os certificados gerados possuísem as seguintes extensões e valores:

- basicConstrains: CA:FALSE
- subjectAltName: email:martins@ufrj.br
- authorityInfoAccess: OCSP; URI:http://icp.ravel.ufjr.br
- crlDistributionPoints: URI: http://icp.ravel.ufrj.br/crl
- nsCertType: server

Após esta configuração, um par de certificados foram gerados, sendo que um deles foi logo revogado para que fosse possível testar tanto a verificação de certificado válido quanto a de certificado revogado.

Para simplificar os testes foram configurados no AirStrikeAP dois domínios virtuais no servidor Apache, cada um deles usando um dos certificados gerados.

A correta operação dos cenários foi verificada ao acessar cada um dos domínios virtuais usando tanto o Internet Explorer como o Netscape. Neste momento o servidor Apache envia o certificado associado ao domínio para o navegador. Este por sua vez obtém a LCR associada ao certificado com base no campo crlDistributionPoints configurado nele. De posse da LCR e do certificado, o navegador executa a verificação.

Caso o certificado esteja válido, a página desejada será apresentada e nenhuma mensagem extra será apresentada, só em caso contrário uma mensagem de erro informando a situação de certificado revogado será apresentada, como na Figura 6-9.

Figura 6-9. Mensagem de erro informando uso de certificado revogado pelo servidor.

A realização do segundo cenário (que pode ser visto como a autenticação dos usuários frente ao servidor usando LCR) foram usadas as mesmas configurações do primeiro cenários.

Para autenticar os usuários junto ao servidor Apache é necessario enviar o certificado do usuário para o servidor. Existem basicamente duas formas de realizar este procedimento, que são:

• Automática: esta forma é possível configurando o Apache de tal modo que ao acessar um dado endereço uma solicitação automática seja gerada pelo servidor ao navegador web pedindo a apresentação de um certificado digital para a conclusão do processo. Este procedimento é feito utilizando a diretiva SSLRequire na configuração Apache.

Essa solitação gerada se apresenta ao usuário na forma de uma janela contendo os certificados pessoais já instalados no programa e solicitando que um deles seja selecionado. Após essa seleção o certificado é enviado e o Apache repassa o mesmo para o mod_ssl que é o responsável por fazer a validação.

Esse procedimento foi exaustivamente estudado e tentado porém devido a falta de documentação sobre o processo não foi possível executá-lo. Essa situação levou a novos estudos e durante a pesquisa foi notado que o processo de autenticação do cliente frente ao servidor não é amplamente utilizado e quando ele é necessário, adota-se a solução clássica dos CGIs, que operam conforme o segundo método, descrito a seguir.

• Manual : este método se vale do processo clássico dos CGIs no qual dados (e arquivos) são enviados ao servidor web via formulários eletrônicos e processados no servidor por programas desenvolvidos específicamente para a função.

O processo ocorre da seguinte forma: o cliente acessa um dado endereço e uma página html com um formulário de envio de arquivo é apresentada. O usuário seleciona o arquivo que deseja enviar e submete o formulário ao programa CGI de tratamento que após realizar sua tarefa informa ao cliente e possívelmente a outros sistemas, o resultado da operação.

O método manual foi escolhido para ser o implementado pois foi possível realizá-lo e analizando este em detalhes verificou-se que ele fornece maior capacidade de customização que o automático. Além disso, o processo de autenticação via OCSP não é suportado pelo par Apache/mod_ssl e neste caso a única solução é a realização do processo usando o método manual (via CGI), o que sendo também feito para a autenticação via LCR torna o procedimento de autenticação independente do mecanismos usado e uniforme.

Outro ponto positivo sobre esta solução é que o usuário não precisa instalar o seu certificado na máquina e/ou programa que estiver usando, podendo assim mantê-lo unicamente em um dispositivo externo. Essa solução evita um problema grave de segurança que advem do fato de que após uma instalação de certificado em uma máquina não-particular o usuário pode esquecer de removê-lo após o uso permitindo que outros usuários usem o seu certificado.

O cenário seguinte a ser realizado (visto como a autenticação do servidor frente ao cliente Netscape usando OCSP) também foi feito com base nas configurações do primeiro cenário.

O navegador web Netscape Communicator, a partir da versão 4.7, possui suporte nativo para o protocolo OCSP podendo teoricamente atuar como cliente neste protocolo e validar certificados junto a um OCSP Responder, como descrito nos capítulos anteriores. Para realizar a autenticação usando este navegador foi necessário apenas configurá-lo de modo a enviar as requisições OCSP de verificação para o endereço do Responder. Todo o resto do processo ocorre do mesmo modo descrito para o processo de validação usando LCRs.

O cenário complementar ao anterior, o de autenticação do cliente Netscape frente ao servidor usando OCSP também foi realizado.

A questão da autenticação do cliente web frente ao servidor Apache via OCSP, devido a falta de suporte no mod_ssl para este processo, só pode ser realizada utilizando CGIs, sendo o procedimento semelhante ao descrito na autenticação do cliente frente ao servidor usando LCRs. A diferença ficou restrita a funcionalidade de cada CGI, que neste caso particular executa o processo de validação do certificado enviado pelo formulário

para o servidor junto a autoridade certificadora usando para isso o protocolo OCSP. Este CGI foi gerado com base no cliente OCSP anteriormente desenvolvido e validado. Este cliente foi devidamente alterado de modo a poder operar como um CGI, devido as diferenças intrínsecas deste mecanismo.

6.3.4 Conclusões

A utilização dos certificados digitais no controle de acesso agrega funcionalidades que tornam o processo como um todo mais rápido e seguro.

O procedimento de validação dos certificados pode ser feito, entre outras formas, utilizando o protocolo OCSP ou o mecanismos das LCRs. Foi possível verificar que o emprego de uma dessas opções não causa diferença significativa na funcionalidade e nem na performance do sistema, abrindo a possibilidade para que um maior número de clientes possam ser usados, não apenas os com suporte ao protocolo OCSP, que são muito raros na atualidade.

Esta capacidade é de extrema importância pois possibilita que aplicações legadas com suporte aos certificados digitais e somente as LCRs (a grande maioria na atualidade) e outras especialmente desenvolvidas para executar o processo de autenticação possam ser usadas neste processo.

Uma vantagem do suporte ao protocolo OCSP pelo cliente é que torna-se possível escalar de forma simples e eficiente os servidores de autenticação. O servidor OCSP implementado pode ser usado independente da autoridade certificadora, reduzindo assim o custo do equipamento usado, a complexidade da configuração e simplificando a gerência e a operação com um todo.

Concluindo, a autoridade certificadora implementada nesta dissertação propicia aos usuários uma forma simples e funcional de obter os seus certificados, tornando o processo simples, do ponto de vista do cliente, devido ao emprego de uma interface web intuitiva e do ponto de vista do gerenciamento pois os administradores poderão de forma independente operar o sistema, ou seja, emitir e revogar os certificados.

A Figura 6-10 apresenta uma proposta de extensão para a arquitetura do AirStrike, mais complexa e funcional que a arquitetura original proposta, porém perfeitamente realizável com o novo ferramental disponível.

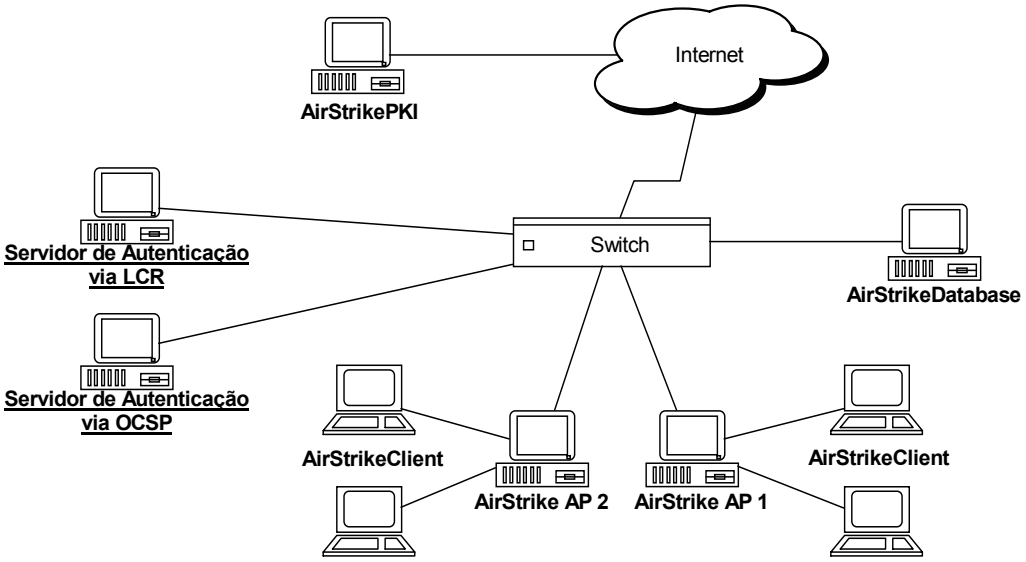


Figura 6-10. Arquitetura estendida do AirStrike

Nessa arquitetura estendida a Autoridade Certificadora fica fora do domínio de validação (delimitado pelo switch) e pode assim atender a vários domínios diferentes. Essa alteração torna necessária a existência de dois outros servidores no domínio de validação, que são: o servidor de autenticação por LCR e o servidor de autenticação por OCSP, necessários para que o processo de validação dos certificados não dependa diretamente da Autoridade Certificadora, tornando o processo como um todo mais rápido.

Com relação as implementações dos cenários, diversos testes foram realizados ao longo do desenvolvimento e ficou claro que existem problemas de compatibilidade dos certificados quando usados no Internet Explorer e no Netscape. Isso foi detectado em várias das combinações de campos de extensão testadas onde verificou-se que o processo de validação do certificado enviado pelo servidor ocorreu com sucesso no Internet Explorer enquanto no Netscape erros não documentados ocorriam.

Além disso, detalhes de configuração podem comprometer o processo de validação chegando ao ponto de não executa-lo, como é o padrão da configuração do Internet Explorer para a verificação dos certificados de servidor enviados. Para que isso ocorra é neces-

sário manualmente alterar essa configuração, que não é de fácil localização. Esse processo pode não ser permitido em alguns sistemas devido aos controles de configurações impostas, inviabilizando a verificação dos certificados.

Outro problema verificado está relacionado com o cache dos resultados das verificações. Em diversas situações após entrar num domínio com um certificado válido e passar para um subdomínio com um certificado revogado esta situação não era informada. A correta situação de revogação só e verificada após reinicializar o navegador e acessando diretamente o subdomínio. Isso demostrou que nesta situação não é possível confiar no resultado de uma verificação feita por um navegador web após o segundo acesso a um site que utilize certificados digitais na criação de um canal seguro.

Com base nessa observações, para obter uma validação confiável dos certificados torna-se necessário ir além dos mecanismos padrões existentes nos navegadores e implementar outros mecanismos que podem ser extensões para estes nevagadores ou adotar clientes dedicados para este fim.

Capítulo 7

Conclusões e Trabalhos Futuros

7.1 Conclusões

Por ser uma linha de pesquisa muito recente, a falta de material e documentação foi percebida durante todo o desenvolvimento dessa dissertação. Apenas para dar uma idéia, foram retornados 111 resultados numa pesquisa feita usando a ferramenta Google com a palavra OCSP, limitada a páginas nacionais, em contra-partida mais de 300 respostas foram encontradas pesquisando LCR+certificado com as mesmas restrições de procura.

Essa falta de recursos dificultou o estudo, ainda mais por esta tecnologia estar sendo usada por mercados altamente lucrativos, onde tudo é segredo de negócio e também porque existe um grave problema relacionado: a padronização.

Ficou claro durante toda a fase de pesquisa e redação desta dissertação que uma dupla concorria a todo momento: de um lado, os organismos que procuravam uma melhor solução, mais genérica; por outro, as empresas ávidas por satizfazer o desejo do mercado por soluções. Este embate chegou a ser tema de estudo pelo IEEE [47].

O que se percebeu com o decorrer do estudo é que não existe um vencedor nessa disputa, os dois lados precisam co-existir. Se todas as empresas gerarem soluções específicas para problemas particulares, não existe o que padronizar. E por outro lado, o que se deve considerar são níveis comuns que, como colocam as recomendações da ITU-T, têm o intuito de uma padronização mínima.

Algumas empresas de ponta nessa tecnologia, como é o caso da RSA Security, contribuem para isso colhendo do mercado as necessidades, detectando os problemas e gerando as soluções. Outros organismos mais rigorosos e com uma estrutura interna mais lenta, acabam ficando com a tarefa de consolidar os padrões.

Com o tempo as soluções, antes inéditas e particulares, acabam tendo suas partes principais padronizadas, o que torna a tecnologia mais sólida, e fornece as empresas de desenvolvimento um diretriz de trabalho e para as empresa de consumo, um guia de escolha onde estas deverão ponderar entre estar no topo da tecnologia, correndo os riscos associados, ou de acordo com os padrões já existentes.

São estas questões que hoje direcionam a solução mais adotada (o padrão PKIX) e também a mais criticada. As questões principais discutidas sobre este modelo giram sobre dois pontos:

- o formato da estrutura de dados que são os certificados digitais: nesse ponto os especialistas advogam que esta estrutura tornou-se complexa demais pois visa fornecer mecanismos para uma gama quase infindável de usos e aplicações. O processo de estruturação do certificado passa por uma fase de abstração necessária para codificar a semântica e os dados em um objeto que precisa ser tratado independentemente da plataforma e da linguagem. Após essa fase, o certificado precisa ser armazenado e transferido entre sistemas distintos e para isso a ITU-T usou outra recomendação chamada de X.209 que não é sempre adotada pois causa uma expansão na quantidade das informações a serem transferidas. Por último e não menos importante, o fato do certificado ser um objeto passivo, ou seja, ele é apenas lido e processado e não possui mecanismos para interferir no próprio processamento, causa uma lacuna com relação aos campos nele existentes. Segundo as especificações alguns campos possuem graus diferentes de tratamento chegando ao limite de impedir o uso do certificado se uma dada condição prescrita no próprio certificado não for entendida pela aplicação, mas a questão é justamente essa, fica a cargo da aplicação tratar esses campos de forma adequada.

- e os modelos de validação: todo certificado passa por três fases, a primeira compreende a localização do certificado, a segunda a verificação e a terceira a validação. As três fases fazem uso da hierarquia de validação para obter os resultados e por isso a própria hierarquia pode tornar-se crítica em alguns casos pois devido ao seu caracter distribuído pode não ser possível ter acesso a algumas partes da árvore. Em outro caso, a perda de confiança em um nó de nível elevado na árvore

pode causar sérios problemas de segurança para todos os níveis dependentes.

Outros problemas advêem das novas semânticas de uso e são iguais para vários modelos de certificados atuais. Alguns desse problemas foram apresentados por um reconhecido especialista em criptografia [48] que foi criticado por dois outros [49][50] mostrando que o tema ainda é bem controverso.

Avançando no desenvolvimento, verificou-se que o procedimento adotado na validação do cliente e servidor para o protocolo OCSP foi demorado, porém eficiente. Não era esperado que na implementação de um protocolo descrito em uma RFC fosse necessário ocorrer uma seqüência de validações dos passos intermediários. Isso foi causado em parte pela escassa documentação existente e em parte pela complexidade oculta tanto do protocolo quanto do emprego das bibliotecas de criptografia. Nos casos de implementação de protocolos semelhantes, é fortemente sugerido que um procedimento de validação de fases, como o empregado, seja adotado. Esta metodologia aumenta gradualmente o domínio sobre o problema sem atrapalhar a evolução do desenvolvimento.

A dificuldade inicial do processo levou a um estudo mais aprofundado de algumas bibliotecas de criptografia (relacionadas no Apêndice II) tornando possível compará-las e verificar que a capacidade das bibliotecas de criptografia existentes para implementar protocolos de segurança depende da maturidade da biblioteca.

Como algumas dessas bibliotecas são bem jovens, e promissoras, é interessante que novas pesquisas relacionadas a implementações que usem criptografia façam um estudo sobre o estado de cada uma das bibliotecas. Acredito que para casos onde apenas os algoritmos de criptografia sejam necessários, não as estruturas para a manipulação de certificados, outras bibliotecas sejam mais apropriadas. Aplicações feitas em outras linguagens, como JAVA terão obrigatoriamente que passar por esta fase pois esta linguagem não foi considerada no estudo realizado.

Com o estudo e a implementação realizados, é possível tecer algumas considerações sobre a segurança e a operação do protocolo OCSP, são elas:

- caso o cliente não consiga obter uma resposta válida de um servidor OCSP, o mesmo precisa implementar uma forma de checagem por LCRs ou dependendo

da situação, utilizar os dois processos para se convencer da resposta obtida.

• o custo computacional das assinaturas pode ser explorado para causar um ataque de negação de serviço (DoS) ao sistema. Uma solução seria pré-computar as assinaturas durante períodos de ociosidade do servidor, por exemplo.

• o fato das mensagens de erro não serem assinadas, alivia a carga do sistema, porém permite que mensagens falsas sejam enviadas aos clientes.

• as requisições não contêm informações sobre servidor do qual está sendo solicitada a situação dos certificados. Isso permite que uma requisição seja enviada para mais de um servidor por um atacante que não possua a autorização de fazer a requisição, com isso se valendo da permissão de outro cliente para obter o resultado desejado.

• todas as implementações estudas até o momento fazem uso do protocolo HTTP para o transporte das mensagens. Implementações eficientes deste protocolo fazem uso intenso de armazenamento temporário (*caching*) o que pode conduzir a problemas de segurança para o cliente, que poderá receber uma resposta inválida presente num cache, assim como para o sistema com um todo, visto que diversas requisições e respostas estarão armazenadas juntas.

• Mesmo sendo um avanço considerável a criação e o uso do protocolo OCSP, este ainda é apenas um protocolo de transporte, ou seja, ele depende de um backend (vide Figura 2-5) que não é definido no protocolo. Por exemplo, se o backend for uma LCR (que é o caso mais geral) o problema da freqüência das informações volta a existir, apenas a sobrecarga do transporte é aliviada (em parte pois o custo da criptografia na geração da requisição e na validação da resposta tendem a ser maior que o processamento das LCRs). Algumas empresas usam bancos de dados de alta performance como backend e armazenam as informações de revogação neles. Neste caso pode-se obter um sistema bem próximo do caso ideal pois não existe o atraso inerente a geração das LCRs.

• O protocolo OCSP, descrito na RFC 2560 é um protocolo para informar

apenas se um certificado foi revogado. É errado pensar que com o OCSP é possível saber se um certificado esta válido ou seja, dentro da sua validade, ou ainda saber se o seu caminho de confiança é válido.

No que tange a inclusão do suporte aos certificados digitais ao ambiente do AirStrike, verificou-se que o ambiente resultante (com a inclusão dos novos cenários de validação dos usuários) tornou-se muito mais funcional e seguro, pois os certificados permitem que seja identificado com exatidão o seu detentor. No caso de ser necessário revogar os certificados, dois processo eficientes e simples existem e toda a administração pode ser centralizada.

Durante o procedimento para a caracterização das LCRs onde ocorreu a coleta de informações sobre as listas de grandes certificadoras, detectou-se nesse processo que algumas empresas não são realmente certificadoras e sim representantes de reais certificadoras. Neste particular, observou-se que a Verisign é onde a maioria das representantes convergem. No Brasil ela é representada pela Certisign e verificou-se que todo certificado emitido pela Certisign fica atrelado a uma LCR da Verisign. Com os resultados obtidos neste procedimento e com as simulações realizadas é possivel concluir que com os controles devidos, o tamanho das LCRs pode ser controlado efetivamente.

7.2 Trabalhos Futuros

As sugestões para trabalhos futuros se dividem em duas classes, uma focada na consolidação da teoria relacionada com a certificação digital e outra que evolui o estudo realizado, que são:

- Embasamento teórico: ficou patente durante todo o período de realização desta dissertação que existe um carência de material sobre o tema. Por este motivo, seria de grande utilidade que trabalhos futuros focassem na geração de pequenas aplicações usando certificados digitais e que estes trabalhos tivessem na documentação um forte componente. E não apenas isso, mas que fossem gerados tutoriais e traduções dos existentes para que este conhecimento pudesse ser rapidamente incorporado ao ambiente de pesquisa acadêmico.

- Evolução dessa pesquisa: a implantação de uma infra-estrutura de chaves

públicas requer que um ambiente físico seguro seja implementado, bem como políticas de segurança associadas aos procedimentos sejam definidas. Como foco para próximas pesquisas seria interessante estudar as políticas de certificação e as declarações de práticas de certificação e implementar ferramentas que auxiliassem na criação, manutenção e verificação dessas políticas. De forma semelhante, seria realmente útil definir e criar mecanismos para que fosse possível verificar a conformidade com os padrões das soluções. Outra sugestão gira em torno da avaliação de novos protocolos de verificação de certificados, como os apresentados em [39] e do estudo detalhado do impacto do uso de certificados digitais no ambiente do AirStrike.

Referências Bibliográficas

[1] NUA Internet: How many online, disponível em http://www.nua.ie/surveys/how_many_online/, endereço visitado em 5 de novembro de 2004

[2] Telecordia Netsizer, disponível em http://www.telcordia.com/research/netsizer/, endereço visitado em 5 de novembro de 2004

[3] Internet Software Consortium, disponível em http://www.isc.org/ds/hosts.html, endereço visitado em 5 de novembro de 2004

[4] Relatórios Web Shoppers, disponível em http://www.webshoppers.com.br, endereço visitado em 5 de novembro de 2004

[5] VIGNA, GIOVANNI. A Topological Characterization of TCP/IP Security, FME 2003: 914-939

[6] BELLOVIN, STEVEN M. Security Problem in the TCP/IP Protocol Suite, Computer Communication Review, 19(2), 1990

[7] BELLOVIN, STEVEN M. Cryptography and Internet, CRYPTO *Conference* (ago 1998)

[8] CARRIÓN, DEMETRIO S. D., e DE MORAES, LUÍS FELIPE M. Implementação de um ponto de acesso para redes 802.11b baseado no OpenBSD, SBRC 2003

[9] BERKOVITS, S., CHOKHANI, S., FURNLONG, J. A., GEITER, J. A. and GUID, J. C. Public Key Infrastructure Study: Final Report, MITRE Corporation, abril de 1994

[10] HOWARD, JOHN D. An Analysis Of Security Incidents On The Internet, 1995, CARNEGIE MELLON UNIVERSITY, disponível em http://www.cert.org/research/JHThesis/Start.html

[11] PERLMAN, R. An overview of PKI trust models. IEEE Network 13, 38-43. 1999.

[12] JUNIOR, JAIME M. DE ALBUQUERQUE, Quais são os profissionais de Tecnologia da Informação? Um Estudo sobre sua característica a partir da oferta de emprego nos últimos trinta anos, Tese de Mestrado, COPPE/UFRJ , março 2003, 193p

[13] CARRIÓN, DEMETRIO S.D. e DE MORAES, LUÍS FELIPE M. AirStrike: Uma Implementação de segurança para redes IEEE 802.11b, apresentado na 1a. Semana de Eletrônica da UFRJ em 2003

[14] POUW, KEEJE D. Segurança na arquitetura TCP/IP: de firewalls a canais seguros, Dissertação de Mestrado, UNICAMP, Janeiro de 1999

[15] BRANCHAUD, MARC. Survey of Public Key Infrastructures, Dissertação de Mestrado, MCGill University, Montreal, Março de 1997

[16] NÉTO, JOÃO CARLOS. Uma implementação do protocolo de micropagamento PayWord, Dissertação de Mestrado, USP, 1999

[17] ARAÚJO, ROBERTO S. DOS SANTOS. Protocolos Criptográficos para Votação Digital, Disertação de Mestrado, UFSC, 2002

[18] PAIXÃO, CESAR A. MONTEIRO, Implementação e Análise Comparativa de quatro variações do criptossistema RSA, Dissertação de Mestrado, IME-USP, em andamento

[19] GUIMARÃES, JOSÉ C. FONTOURA. Proposta de fortalecimento do sistema criptográfico DES contra criptanálise diferencial, Projeto Final, USP, 1993

[20] HONDA, MARIÊ IKEMOTO, Cifra autodecriptável com o algoritmos Blowfish, Projeto Final, Departamento de Ciência da Computação, UnB, junho de 1998.

[21] PEREIRA, FABIANO CASTRO. Ostracom: Votação Digital Segura pela Internet, Disertação de Mestrado, UFSC, em andamento

[22] KAZIENKO, JULIANO FONTOURA. Assinatura Digital de Documentos Eletrônicos através da Impressão Digital, Dissertação de Mestrado, UFSC, 2003

[23] DEMÉTRIO, DENISE BENDO, Infra-estrutura para Protocolização Digital de Documentos Eletrônicos, UFSC, 2003, 140p

[24] PASQUAL, EVERTON SCHONARDIE, IDDE - Uma Infra-estrutura para a Datação de Documentos Eletrônicos. 2002 110p dissertacao de mestrado, UFSC

[25] BORISOV, N., GOLDBERG, I. and WAGNER D. Intercepting Mobile Communications: The insecurity of 802.11, 7th Annual Internation Conference on Mobile Computing and Networking.

[26] CASOLE, M., WLAN Security - Status, Problems and Perspective, European Wireless 2002

[27] CALLAS, J., DONNERHACKE, L., FINNEY, H., THAYER, R., RFC 2440 OpenPGP Message Format. November 1998.

[28] DIFFIE, W., HELLMAN, M.E, New Directions on Cryptography, IEEE Trans. on Information Theory, Vol. IT-22, No 6, Novembro de 1976

[29] KOHNFELDER, Loren M. Towarda a Practical Public-key Cryptosystem, B.Sc. thesis, MIT Department of Electrical Engineering, Maio de 1978

[30] RIVEST, R.L, SHAMIR, A. e ADLEMAN, L., A Method for Obtaining Digital Signatures and Public-key Cryptosystems, Communications of the ACM, Vol 21, Fevereiro de 1978

[31] ITU-T Recommendation X.501 (2001) | ISO/IEC 9594-2:2001, Information technology . Open Systems Interconnection . The Directory: Models

[32] ITU-T Recommendation X.208 (1994) | ISO/IEC 8824-1:1994, Specification of Abstract Syntax Notation One (ASN.1)

[33] ITU-T Recommendation X.209 (1994) | ISO/IEC 8825-1:1994, Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)

[34] ITU-T Recomendation X.509 (1997 E) : Information Technology - Open System Interconnection - The Directory: Authentication Framework, Junho de 1997

[35] KENT, S. Privacy Enhancement for Internet Electronic Mail, RFC 1422, Fevereiro de 1993

[36] HOUSLEY, R., POLK, W., FORD, W., SOLO, D., Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC3280), abril de 2002

[37] KOCHER, P.C, On certificate revocation and validation, International Conference on Financial Cryptography, fevereiro de 1998

[38] MERKLE, R.C., A certified digital signature, Advances in Cryptology, Lecture Notes in Computer Science, número 435, 1989

[39] ARSENAULT, A., TURNER, S. Internet X.509 Public Key Infrastructure: Roadmap, PKIX Working Group, julho de 2002

[40] KON, FÁBIO. O software aberto e a questão social, Relatório Técnico RT-MAC-2001-07, IME, USP

[41] Linux Today: http://linuxtoday.com/news_story.php3?ltsn=2002-08-30-011-26-NW-LL-PB - endereço visitado em 5 de novembro de 2004

[42] Linux Today: http://linuxtoday.com/news_story.php3?ltsn=2002-08-30-011-26-NW-LL-PB - endereço visitado em 5 de novembro de 2004

[43] ZDNet: http://www.zdnet.com/zdnn/stories/news/0,4586,2779806,00.html - endereço visitado em 5 de novembro de 2004

[44] EWeek: http://www.eweek.com/article2/0,3959,562220,00.asp - endereço visitado em 5 de novembro de 2004

[45] EWeekhttp://www.eweek.com/article2/0,3959,562226,00.asp - endereço visitado em 5 de novembro de 2004

[46] Research Labs: http://www.research.avayalabs.com/techabstractY.html#ALR-2002-003 - endereço visitado em 5 de novembro de 2004

[47] SHERIF, M.H. Standardization and Innovation in Information Technology, IEEE Communications, janeiro de 2000

[48] ELLISON, C., SCHNEIER, B., Ten Risks of PKI: What you’re not being told about public key infrastructure, Computer Security Journal, Volume XVI, Número1, 2000

[49] LAURIE, BEN. Seven and a Half Non-risks of PKI: What You Shouldn't Be Told about Public Key Infrastructure, disponivel em http://www.apache-ssl.org/7.5things.txt - endereço visitado em 5 de novembro de 2004

[50] PEREZ, A. A Response to 10 Risks of PKI, disponível http://home.pacbell.net/aram/responsetenrisks.html

[51] KERCHHOF, AUGUSTE, La cryptographie militaire, Journal des sciences militaires, vol. IX, pp. 5–38, Jan. 1883, pp. 161–191, Feb. 1883. disponível em http://www.cl.cam.ac.uk/~fapp2/kerckhoffs/#english

[52] KAHN, DAVID. The Codebreakers: The story of secret writing, Macmilian Publishing Co., New York, 1967

[53] SAMPLE, M. Snacc: A High Performance ASN.1 to C/C++ Compiler, Department of Computer Science, University of British Columbia, July 1993

[54] OASIS, Ferramenta para autenticação centralizada, disponível em http://software.stockholmopem.net, site acessado em 10/02/2004

[55] NoCat, Ferramenta para autenticação centralizada, disponível em http://www.nocat.org, site acessado em 10/02/2004

[56] Netlogon, Ferramenta para autenticação centralizada, disponível em http://www.unit.liu.se/dokument/natverk/netlogon.html, site acessado em 10/02/2004

[57] WALKER, JESSE R. Unsafe at any key size: An Analysis of the WEP encapsulation, IEEE Document 802.11-00/362

Apêndice I: Conceitos Básicos sobre Criptografia

O objetivo deste apêndice é apresentar algumas das técnicas de criptografia e os serviços por ela oferecidos, em especial quando empregados na segurança de uma comunicação. Nele será abordada a adequação das duas formas existentes de criptografia para uso na Internet e também um procedimento genérico para assinatura digital.

O capítulo apresenta algumas definições importantes para a compreensão do assunto, aborda os dois tipos de criptografia existentes e suas aplicações, como por exemplo nas funções resumo e na assinatura digital.

I.1 Definições Iniciais

Esta seção tem o objetivo de agrupar algumas definições que serão empregas no restante deste texto. Isto torna-se necessário pois, devido a origem inglesa de muitos dos termos, a tradução resultante acaba criando expressões que não pertencem ao nosso vocabulário ou estrangeirismos desnecessários, já que o português é um idioma riquíssimo.

Os principais termos que necessitam ser definidos são os seguintes:

- Confidencialidade:

É o serviço usado para manter o conteúdo da informação protegido, de forma a ser acessado ou entendido apenas por pessoas autorizadas.
Existem várias formas de ser implementada, como por exemplo, através de meios físicos ou por algoritmos matemáticos.

- Integridade dos dados:

É o serviço que garante que os dados não serão alterados. Para ser garantido, são necessários mecanismos capazes de detectar a manipulação dos dados. A manipu-

lação dos dados inclui inserção, remoção e substituição.

• Autenticidade:

É um serviço relacionado com a identificação. Se aplica tanto à informação em si quanto às partes que participam de uma comunicação.

Na criptografia, essas duas vertentes são chamadas de autenticação da entidade e autenticação da origem dos dados; esta por sua vez está intimamente relacionada com a integridade dos dados.

• Não Repúdio ou Irretratabilidade:

É a incapacidade do emissor, após ter executado um dado processo, negar a propriedade de tal ato.

• Emissor, Receptor e Atacante:

Em quase toda a literatura sobre o tema, é comum se referenciar a entidade que origina a mensagem com Alice (devido a letra A) e ao receptor com Bob, caso um terceiro participante seja necessário, ele é chamado Charlie. O papel de atacante fica a cargo de Oscar.

• Mensagem, codificação, cifragem e criptograma:

A mensagem (ou texto plano) é o material a ser enviado por Alice. Nas referências em inglês, usa-se o termo *plaintext*.

O termo codificação é muitas vezes usado (e confundido) com cifragem. Para resolver a dúvida, codificar é escrever código, como programar em C ou HTML.

Em contra partida, cifrar é escrever em cifra, o sentido é ocultar algo através do método empregado. Em inglês usa-se os termo *encryption/decryption* ou *encipher/decipher*. Na tradução, teríamos algo como encriptar/decriptar. Pelo dicionário Aurélio encriptar é "colocar na cripta" (decriptar não existe), por isso fica melhor usar cifrar e decifrar.

O resultado do processo de cifragem é o criptograma ou do termo inglês, *ciphertext*.

• Algoritmos e Chaves:

Um algoritmo criptográfico (também chamado cifra) é uma função matemática usada para cifrar e decifrar.

Antigamente os algoritmos carregavam junto com o método todo o segredo do processo e por isso não podiam ser revelados, atualmente, separa-se o segredo (a chave) do algoritmo em si (essa separação é conhecida como princípio de Kerckhoff[51]), trazendo grandes vantagens com isso, como por exemplo, a capacidade de se avaliar o funcionamento do algoritmos sem comprometer a sua segurança.

Associada a chave (*key*) existe o termo *keyspace* que se refere ao espaço de possíveis chaves a serem usadas no processo. Este espaço por sua vez é limitado normalmente pelo tamanho de bits da chave.

I.2 Algoritmos de Criptografia

A criptografia[11], que é um dos ramos da criptologia, é a ciência que cuida de estudar os processos de cifragem e decifragem para fornecer um determinado nível de segurança aos dados, de modo que os mesmos possam ser armazenados, transmitidos e recuperados sem sua alteração, criação ou exposição não autorizada. Em outras palavras, técnicas de criptografia podem ser usadas como um meio efetivo de proteção de informações suscetíveis a ataques, estejam elas armazenadas em um computador ou sendo transmitidas pela rede. Seu principal objetivo é prover uma comunicação segura[12], garantindo serviços básicos de autenticação, privacidade e integridade dos dados. Sua origem é muito antiga, muitos antes do homem imaginar a máquina de escrever os egípcios já usavam técnicas de criptografia, isso apriximadamente em 4000 A.C. Grande parte da sua rica evolução pode ser lida no texto de D. Kahn[52].

Com o advento dos computadores de grande capacidade de processamento, as funções criptográficas ficam cada vez mais complexas.

Os algoritmos, e suas chaves, evoluíram e recentemente dois grupos principais podem ser caracterizados: os algoritmos simétricos e os assimétricos. O texto que segue trata algumas questões que servem de base para uma explicação mais detalhada dos pro-

11. A palavra criptografia tem origem grega (kriptos = escondido e grifo = grafia) e define a arte ou ciência de escrever em cifras, na sua fase inicial, usando artifícios lúdicos e recentente, usando técnicas matemáticas.

12. No caso do armazenamento, a comunicação pode ser vista como sendo do usuário para si mesmo.

blemas de segurança sobre o foco desses dois ramos e aborda as razões pelas quais a criptografia simétrica é pouco adequada para a operação na Internet e apresenta o melhor resultado obtido com o emprego das chaves assimétricas e dos certificados digitais.

I.2.1 Criptografia Simétrica

Na comunicação, uma necessidade básica de segurança é garantir que as mensagens sejam privadas. Por privacidade, entende-se que, ao transitar pela Internet, a mensagem não poderá ser compreendida por uma terceira entidade, que não seja autorizada, considerando que apenas Alice e Bob têm direito de acesso sobre a mensagem. Note que esta terceira entidade pode ter acesso aos bits da mensagem, mas para esta, o conteúdo fica inacessível (vide Figura I-1).

Figura I-1. Modelo simplificado de comunicação usando criptografia

Para proteger essa mensagem, garantindo a privacidade, ou confidencialidade, um primeiro ítem da solução de segurança é o emprego, por Alice e por Bob, de um algoritmo de criptografia capaz de transformar a mensagem original em uma mensagem cifrada, ou seja, não compreensível por uma terceira entidade.

A criptografia simétrica requer que o transmissor e o receptor compartilhem uma mesma chave. Essa informação secreta deve ser usada para cifrar e decifrar as mensagens. Se a chave permanecer secreta, então ninguém mais, além do transmissor e do receptor em questão, poderá ler a mensagem. Se Alice e o Bob conhecem a mesma chave secreta e não a compartilham com mais ninguém, então eles poderão enviar mensagens privadas

com a certeza de que ninguém mais será capaz de obter o conteúdo da mensagem. Esquematicamente temos:

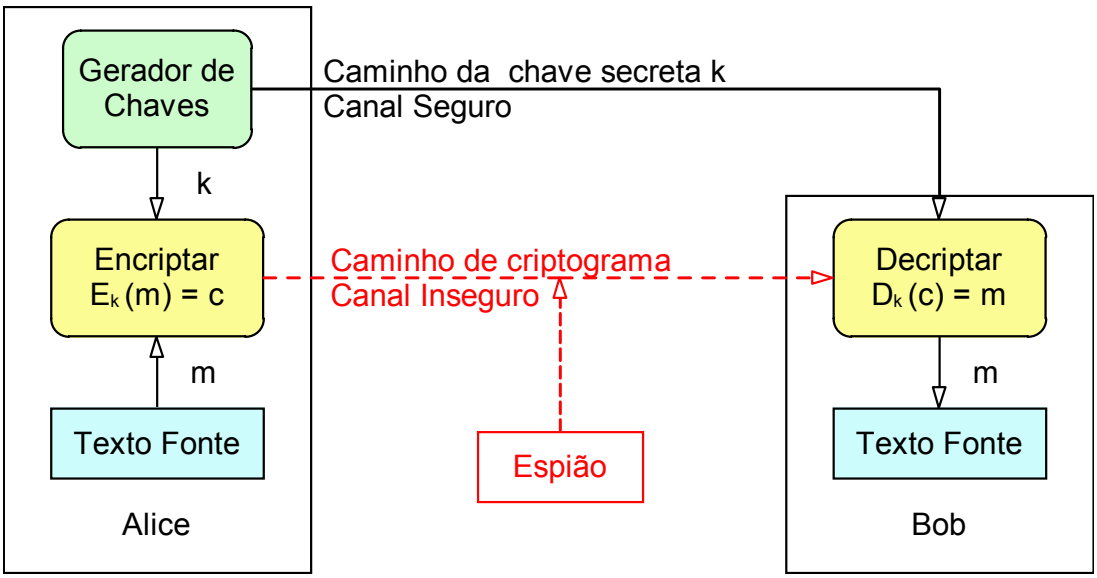


Figura I-2. Esquema de comunicação usando a Criptografia Simétrica

I.2.2 Criptografia Assimétrica

Também conhecida como criptografia de chave pública, nela são empregados dois algoritmos que usam cada um uma chave diferente, uma delas dita pública e a outra privada, que são utilizadas para cifrar, decifrar e em alguns casos assinar a mensagem. Se uma das chaves é usada para cifrar uma mensagem, então a outra deve ser usada para decifrar. Assim o Bob gera duas chaves, guarda uma delas, a denominada chave privada, e torna pública a outra. Deste modo, Alice pode cifrar uma mensagem usando a chave pública do Bob obtida e enviar para este a mensagem cifrada pois apenas ele, o Bob, que é o detentor da chave privada, poderá decifrar a mensagem enviada.

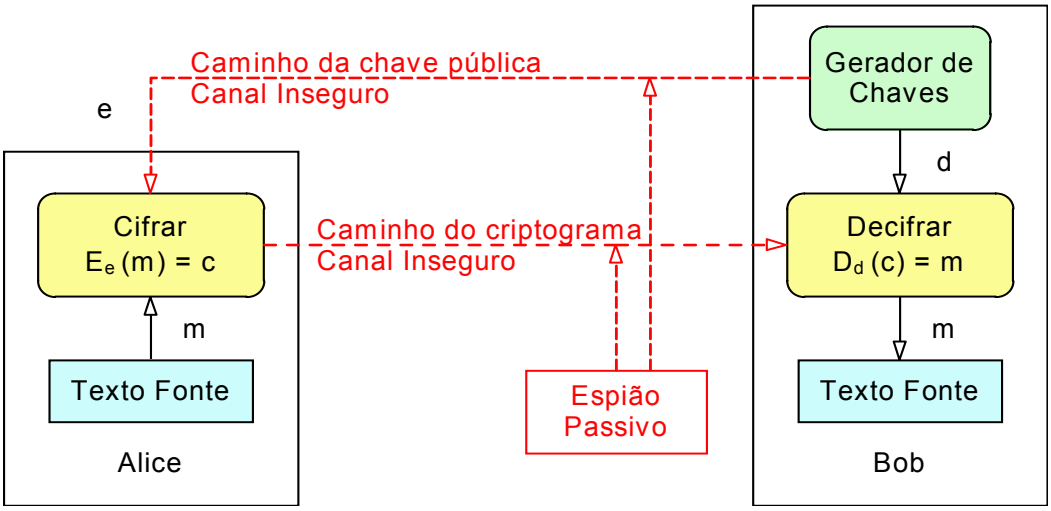


Figura I-3. Esquema de comunicação usando a Criptografia Asssimétrica

A diferença nos dois processos, como se pode notar pela Figura I-3, está no par algoritmo-chave. No caso simétrico, a chave é única[13] e o algoritmo opera em um sentido para cifrar e no sentido inverso para decifrar. No caso assimétrico, são duas as chaves e são dois os algoritmos. As chaves são geradas por processos matemáticos que garantem ser computacionalmente difícil dada uma, calcular a outra.

A utilização do sistema de criptografia convencional (ou simétrica) requer que uma chave secreta seja compartilhada entre cada par de entidades com necessidades de comunicação, o que implica na necessidade de uma quantidade de chaves correspondente ao quadrado[14] da quantidade de entidades comunicantes. Uma grande quantidade de chaves acarreta sérios problemas de distribuição, proteção e manutenção, tornando a criptografia simétrica menos adequada para o ambiente *web* do que a com chave pública, onde é suficiente que cada entidade guarde, proteja e dê manutenção apenas à sua própria chave privada, divulgando amplamente a correspondente chave pública. Por essa razão, ainda que na *web* seja possível a utilização de criptografia simétrica e que essa forma de criptografia seja efetivamente utilizada em determinados momentos da comunicação, a criptografia com chave pública se adeqüa melhor.

13. Para ser extato, a definição mais aceita diz que na criptografia simétrica as chaves não precisam ser iguais, mas de posse de uma a outra é facilmente deduzida (Handbook of Applied Cryptography, pg 15).
14. Matematicamente, temos que $\lim_{n \to \infty} \frac{n(n-1)}{2} = n^2$

I.3 Aplicações

I.3.1 Funções de *Hash*

Ainda que Alice possa cifrar sua mensagem tornando-a secreta, há outros riscos a serem considerados. Por exemplo, uma terceira entidade (espião) pode interceptar uma série de mensagens enviadas por Alice, e de posse destas, montar (manipulando o material obtido) uma mensagem, sem para tal conhecer a chave usada. A questão de segurança nesse caso é a necessidade de garantir que as mensagens não sejam modificadas, em parte ou no todo, ao serem transferidas, ou seja, trata-se da necessidade de garantir a integridade da mensagem.

A maneira de Alice garantir a integridade da mensagem é calculando um resumo (chamado *hash value)* da mensagem e enviá-lo ao Bob junto à própria mensagem. O receptor da mensagem, ao recebê-la, executa o mesmo cálculo, comparando este com o enviado por Alice. Se os dois resumos forem iguais, considera-se que a mensagem foi recebida sem alterações.

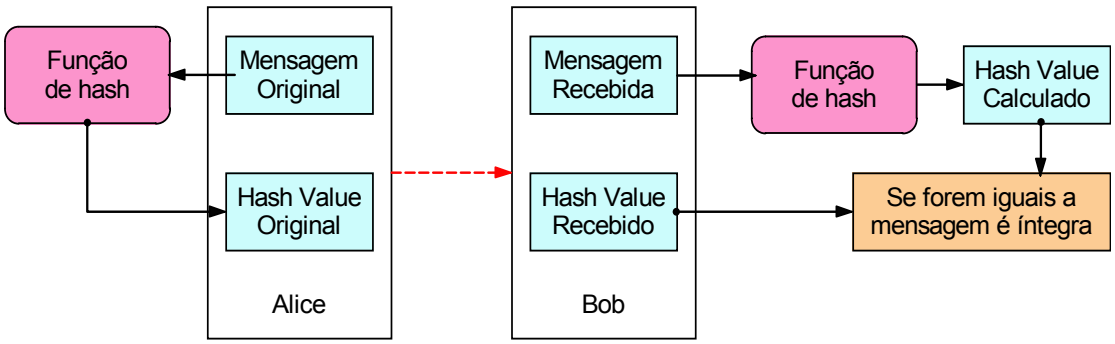


Figura I-4. Processo de validação de integridade de uma mensagem

O resumo (*hash value*) da mensagem é calculado usando uma classe de funções matemáticas chamadas de *Hash Functions* que são geralmente derivadas das funções utilizadas na criptografia simétrica. Os geradores de resumo[15] de mensagens são usados para obter como resultado uma representação de tamanho fixo e bem pequena comparada com as mensagens, que podem ter tamanho imprevisivelmente grande. A probabilidade de se

15. Na literatura nacional, em muitos casos usa-se o termo digestor de mensagem.

encontrar duas mensagens que sejam reduzidas ao mesmo *hash value* é muito pequena e depende do tamanho adotado para esse *hash*.

O problema na utilização dos resumos, é que Alice deve conseguir uma maneira de enviar cada resumo para o Bob seguramente, isto é, sem que o próprio resumo possa ser alterado por uma terceira entidade ao percorrer a Internet; quando isto é alcançado, a integridade da mensagem é garantida. Para proteger o próprio resumo, é necessário mais um elemento de proteção: a assinatura digital.

I.3.2 Assinaturas Digitais

Além da questão da integridade da mensagem, a assinatura digital é empregada também para suprir uma outra necessidade: a garantia por parte do receptor, da origem (quem foi o remetente) da mensagem. Quando Alice envia sua mensagem para o Bob, este necessita da garantia de que a mensagem realmente partiu de Alice, ou seja, que a mensagem além de íntegra e privada, é autêntica e que a operação pode efetivamente ser realizada em nome de Alice.

A assinatura digital, criada por Alice e incluída na mensagem, consiste do cálculo de um resumo da mensagem por uma função de *hash*, e da cifragem deste resumo, juntamente com outras informações, tal como um número de série, usando um algoritmo de chave pública, mas realizando a operação criptográfica usando a chave privada do transmissor. Como qualquer entidade pode decifrar o resumo cifrado[16], bastando para tanto usar a chave pública do transmissor e, como apenas o transmissor-assinante conhece a chave privada correspondente, há uma efetiva garantia de que apenas ele, o transmissor, pode ter assinado a mensagem. Essa garantia reside no fato da chave privada ser apenas do conhecimento da entidade que assina a mensagem. A Figura I-5 resume o processo.

16. O fato de decifrar o *hash value*, usando a chave pública, que foi criptografado com a chave privada pode ser chamado de validação de assinatura.

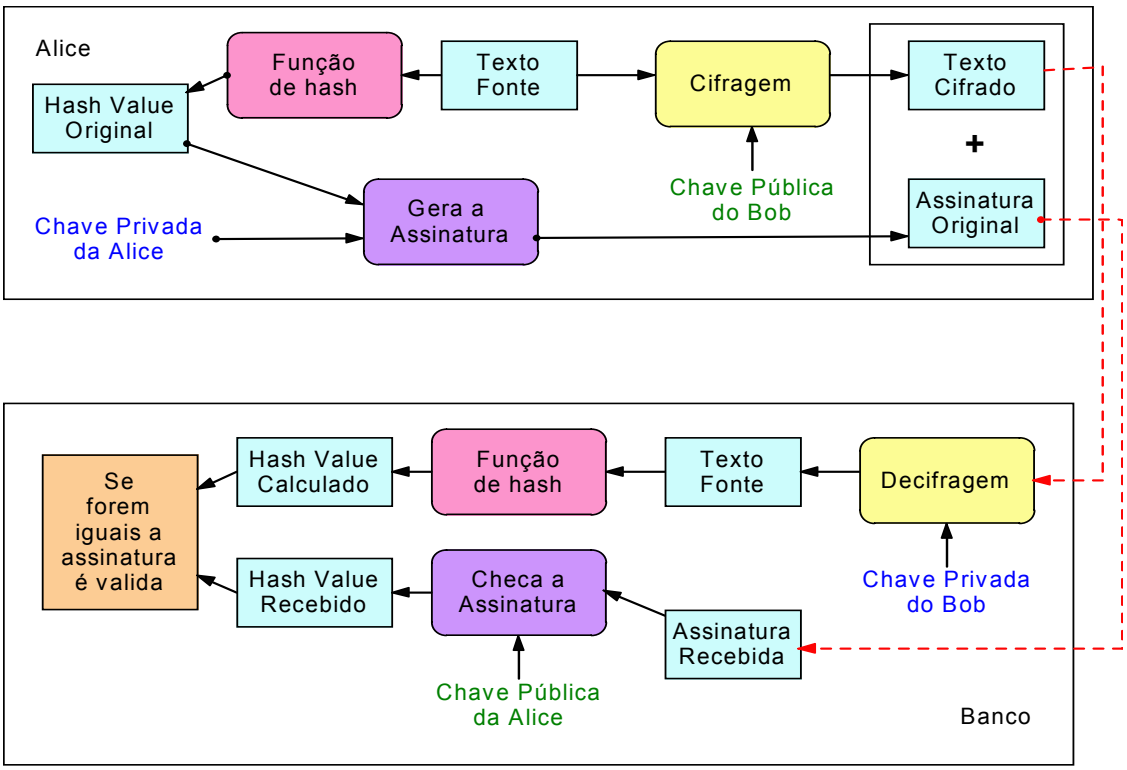


Figura I-5. Esquena de assinatura eletrônica

Além disso, visto que o resumo só pode ser cifrado usando a chave privada do transmissor, uma terceira entidade, que não conhece tal chave, não poderá modificar a mensagem sem ser detectado pois não terá como gerar um resumo cifrado correspondente à mensagem alterada.

Desse modo, com a assinatura digital e a presença de um resumo válido, há uma garantia da integridade da origem da mensagem.

A premissa assumida para afirmar que uma mensagem foi assinada por um pessoa recai sobre outro problema, o relacionamento entre entidades sem a possibilidade de contato físico. Quando uma entidade gera o seu par de chaves e diponibiliza a pública, ela pode se dizer ser quem ela desejar. A associação nome *x* chave pública é feita pelo próprio criador da chave. É exatamente este o ponto ainda frágil do processo apresentado até aqui. Quando é possível um contato pessoal, os participantes podem se identificar fisicamente e trocar as chaves públicas, como se fossem cartões de visita e repassar para outros estas chaves. Mas isto não é de forma alguma realizável quando pensamos em todos os usuários ou dispositivos atuanto na rede pois: A) pode não ser possível um encontro presencial e

B) dispositivos não possuem rostos ! É para resolver esta questão que foram criados os certificados digitais, eles são emitidos por entidades que gozam da confiança das partes e são responsáveis por todo o processo de geração das chaves e associação com a entidade.

I.3.3 Certificado Digital

Diffie e Hellman num inspirador artigo sobre criptografia de chave pública[28], indicaram o problema do estabelecimento da autenticidade do par nome-chave, mencionado anteriormente. A sugestão dada por eles foi o emprego de repositórios online seguros com entradas mapeando o par nome-chave. Louren M. Kohnfelder, que compartilhava da mesma idéia, menciona em sua tese[29] o mesmo problema.

Em 1978[17] ela propôs, para solucionar uma questão de performance e segurança detectada por Diffie e Hellman na validação das associações, a criação de uma estrutura de dados assinada contendo o nome e a chave pública. A esta estrutura ela chamou simplesmente de certificado digital, mas atualmente são referênciados como certificados de chave pública ou PKC (*Public Key Certificate*). Por ser a estrutura assinada, ela poderia ser passada por entidades não confiáveis resolvendo também um outro problema, o da centralização da distribuição dos certificados. Na proposta da Kohnfelder, poucas informações foram dadas sobre como seria a estrutura destes certificados e muito menos como seria o seu ciclo de vida, pontos de grande importância na atualidade.

Essa idéia permaneceu por um bom tempo adormecida até que para resolver um outro problema, o da disponibilização de informações estruturadas em larga escala, a ITU-T despertou a idéia dos certificados digitais.

A idealização dos certificados passou por um processo de concretização complexo, dado seu amplo intento original pela ITU-T. Algumas fase vitais foram:

- a padronização da forma como os dados seriam descritos no certificado, para tal a ITU-T lançou mão da sintaxe chamada *Abstract Syntax Notation One* - ASN.1[32], que introduz o conceito de *Object Identifieres* (OIDs) e torna possível a comunicação da semântica da informação entre sistemas distintos. Todo objeto

17. Nesse mesmo ano, o artigo de Rivest, Shamir e Adleman [30] foi publicado, mostrando de fato com poderiam ser implentadas as idéias de Diffie e Hellman.

interno a um certificado digital possui um OID associado a ele.

• a padronização dos próprios certificados (seu conteúdo e sua estrutura interna) que primeiramente foi feita na recomendação X.509[34]

• como originalmente os certificados eram de identidade, havia a necessidade de padronizar a forma como os nomes seriam hierarquicamente organizados neste sistema, isso foi feito com o auxílio da recomendação X.500[31] que além disso também estabelece como seria possível localizar, recuperar, inserir e remover qualquer tipo de infomação de forma consistente e distribuída, conhecido como serviço de diretórios e introduz os conceitos de *Distinguished Name* e *Relative Distinguished Name*.

• para efeito de armazenamento, convencionou-se que os dados seriam codificados segundo um outro padrão, o *Distinguished Encoding Rules* (DER)[33] de modo a poderem ser armazenados e transferidos indenpendentemente da plataformade hardware e software.

É provável que essas recomendações tenham sofrido a influência da proposta da Kohnfelder devido as semelhanças nas abordagens. Essa recomendação (a X.509) fornece a base para a atual implementação da mais importante vertente de certificados, o padrão PKIX.

A Figura I-6 mostra um exemplo de uma hierarquia de nome X.500 que também é utilizada para organizar os OID dos objetos presentes nos certificados.

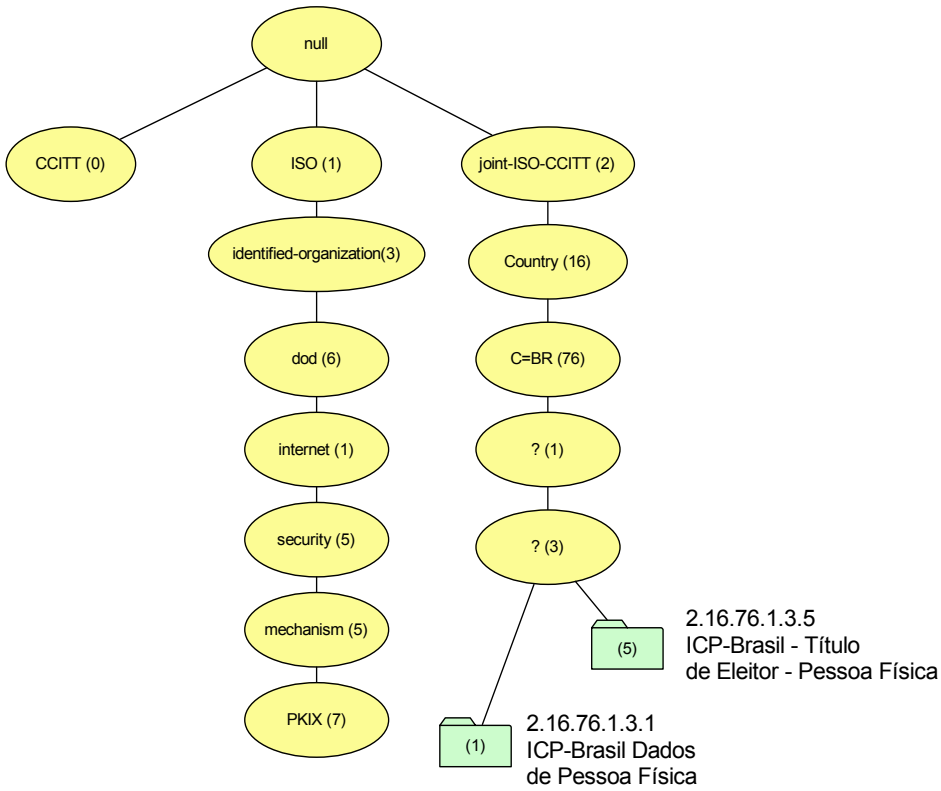


Figura I-6. Exemplo de hirarquia segundo o padrão X.500

Os certificados criados para prover a segurança necessária à recomendação X.500 serviram de ponto de partida para os atualmente utilizados. A primeira aplicação desses certificados fora do padrão X.500 foi no *Privacy Enhanced Mail* (PEM)[35] em 1993. As versões dos certificados X.509 foram evoluindo segundo as necessidades e as constatações de que os modelos de segurança originais não se adeqüavam as exigências.

O padrão possui dois formatos de certificados, um para gerar a associação da entidade-fim (no original, *End Endity*) com a respectiva chave pública, chamado apenas de certificado e um outro, bem semelhante, para divulgar uma lista dos certificados que foram revogados, chamados de lista de certificados de revogados.

Os certificados, segundo a recomendação X.509, passaram por 3 versões, graficamente representadas na Figura I-7.

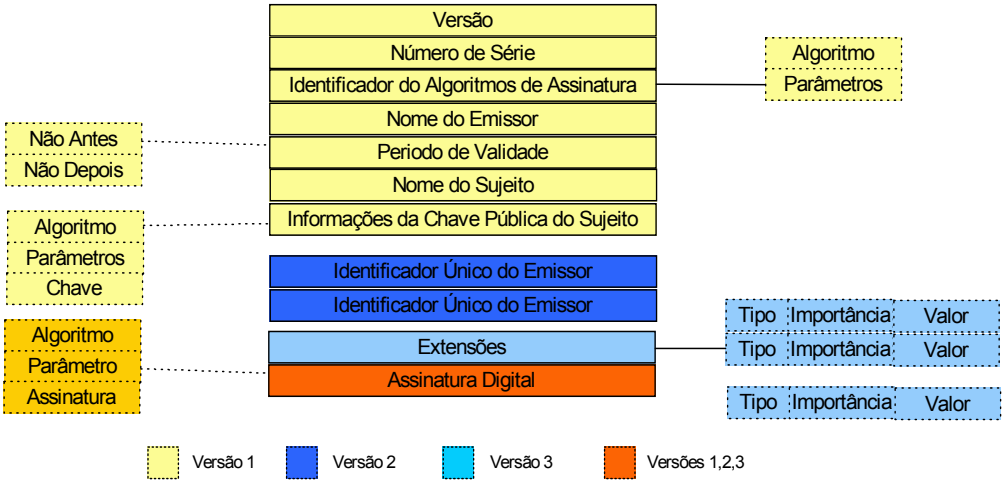


Figura I-7. Versões da estrutura de um certificado digital X.509.

As razões para as mudanças são resumidas a seguir:

• X.509v1: tinha um número pequeno de campos o que limitava sua utilização. Além disso, alguns problemas de segurança foram identificados no padrão. Foi o adotado pelo padrão PEM em 1993.

• X.509v2: por ocasião da revisão do protocolo X.500 a primeira versão foi totalmente revisada. Nesta versão foram adicionados novos campos (vide Figura I-7) com o objetivo de possibilitar a reutilização de nomes iguais em diferentes certificados digitais.

• X.509v3: foi detectado, com a experiência obtida na tentativa de implementar o PEM que os formatos da versão 1 e 2 eram deficientes em vários aspectos. Em resposta a isso, na atual versão, de junho de 1997, foram adicionados campos de extensão, os quais tornma o certificado mais flexível e com um leque de utilização muito maior.

Embora existam alguns outros formatos de certificados em uso na Internet, sem dúvida a descrita na recomendação X.509 e padronizada pelo IETF como PKIX é a mais aceita por vários motivos, muitos dos quais serão abordados no decorrer desta dissertação.

Estruturalmente, um certificado digital é um arquivo de dados, subdivididos em seções contendo informações obrigatórias e extensões de processamento condicional.

I.3.3.1 Lista de Certificados Revogados

As Listas de Certificados Revogados, ou simplismente LCRs, armazenam uma relação de certificados revogados, da forma apresentada na figura abaixo.

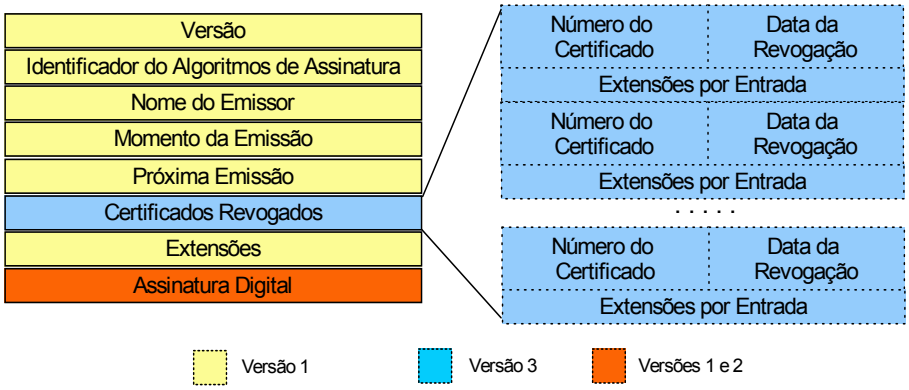


Figura I-8. Layout da versão 2 da CRL do X.509

Neste certificado, a semântica dos campos é a seguinte:

• Versão

Este campo identifica qual versão do formato de revogação se aplica ao certificado.

• Identificador do Algoritmo de Assinatura

Este campo identifica o algoritmo e os parâmetros usados pela entidade ao assinar o certificado;

• Nome do Emissor

Contém o *Distinguished Name* da entidade que assina e divulga o certificado;

• Momento da Emissão

Este campo contém o dia e hora da promulgação do certificado.

• Próxima da Emissão

Este campo contém o dia e hora da próxima promulgação do certificado de revogação. Caso a frequência de promulgação seja conhecida por todos, este campo pode ser omitido. O padrão recomenda sempre a sua presença e é possível promulgar um outro certificado antes da data assinalada, mas nunca após.

• Certificados Revogados

Armazena três informações: o identificador do certificado revogado, a data efetiva da revogação e algumas extensões (apenas na versão 2) que podem armazenar, por exemplo, o motivo da revogação.

- Extensões

Existentes apenas na versão 2 do formato e reservam um local para que novos campos possam ser adicionados, no mesmo molde do certificado da Figura I-7. Estas extensões podem se aplicam a uma entrada em particular ou a todo o certificado.

Apêndice II: Bibliotecas de Criptografia

Este apêndice resume as informações mais importantes e utilizadas na análise das bibliotecas de criptografia visando uma seleção para a utilização na implementação do cliente e do servidor OCSP apresentado no capítulo 5.

II.1 Biblioteca de ASN.1

Para a manipulação de objetos ASN.1 foi avaliada a biblioteca SNACC[53]. Essa biblioteca foi considerada para avaliação durante uma fase inicial de estudos na qual esperava-se que as implementações a serem realizadas teriam que ser feitas partindo da implementação ASN.1 das mensagens. Depois de algumas avaliações, percebeu-se que essa abordagem teria um alto grau de complexidade desnecessário pois outras bibliotecas que foram sendo analisadas mostraram que seria possível lidar com os certificados em um nível mais alto de abstração.

II.2 Bibliotecas para Criptografia

A relação apresentada nesta seção é limitada as bibliotecas mais maduras, ou seja, àquelas que possuem um grupo de desenvolvimento ativo, uma boa documentação, exemplos de uso bem diversificados e um grupo de usuários também ativo. Outras bibliotecas jovens que ficam de fora desse crivo existem, mas foram desconsideradas nesta avaliação.

As bibliotecas estudadas foram:

- Criptlib

Autor: Peter Gutman

Homepage: http://www.cs.auckland.ac.nz/~pgut001/cryptlib/

Linguagem: C

Plataformas: x86 (Linux, Windows 16/32bits, DOS), OS/2, BeOS, Macintosh, VM/CMS e MVS

• Personal Security Manager (PSM)

Autor: Netscape

Homepage: http://www.mozilla.org/projects/security/pki/psm/

Linguagem: C

Plataformas: x86 (Linux, Windows 16/32bits, DOS), OS/2, BeOS, Macintosh

• Network Security Services (NSS)

Autor: Netscape

Homepage: http://www.mozilla.org/projects/security/pki/nss/index.html

Linguagem: C

Plataformas: x86 (Linux, Windows 16/32bits, DOS), OS/2, BeOS, Macintosh, VM/CMS e MVS

• Crypto++ 5.1

Autor: Wei Dai

Homepage: http://www.eskimo.com/~weidai/cryptlib.html

Linguagem: C++

Plataformas: x86 (Linux, Windows e BeOS), SPARC (Linux e Solaris), PowerPC (MacOS X)

• Botan

Autor: Jack Lloyd

Homepage: http://opencl.randombit.net/

Linguagem: C++

Plataformas: x86 (Linux e Windows), SPARC (Linux e Solaris), PowerPC (MacOS X) e Alpha (Tru64) e MIPS (Irix)

Apêndice III: Certificados e Programas

Este apêndice se divide em duas partes: uma dedicada a apresentar o conteúdo de alguns certificados utilizados durante o processo de estudos e outra para apresentar os scripts criados para auxiliar nas tarefas de simulação e geração dos gráficos.

III.1 Certificados Utilizados

III.1.1 Banco do Brasil

Este certificado foi obtido diretamente do site do Banco do Brasil no endereço www.bb.com.br. O conteúdo do certificado é apresentado a seguir.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            63:ae:a3:34:9d:26:4e:30:44:af:3a:3c:68:d0:12:c3
        Signature Algorithm: md5WithRSAEncryption
        Issuer: O=VeriSign Trust Network, OU=VeriSign, Inc., OU=VeriSign Inter-
national Server CA - Class 3, OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY
LTD.(c)97 VeriSign
        Validity
            Not Before: Aug 30 00:00:00 2003 GMT
            Not After : Aug 29 23:59:59 2004 GMT
       Subject: C=BR, ST=Distrito Federal, L=Brasilia, O=Banco do Brasil S.A.,
OU=DITEC-GETEC, OU=Terms of use at www.verisign.com/rpa (c)00, CN=cert.bancodo-
brasil.com.br
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:d0:d6:a5:e1:89:f8:da:fb:ca:71:03:0c:c8:3f:
                    08:2c:15:13:b5:59:a7:3b:04:0a:bc:7a:18:ab:3d:
                    42:7e:52:a4:9b:ab:ae:3e:16:32:b6:c3:b7:d3:03:
                    ae:25:02:ec:2d:c5:a7:e5:e3:ba:27:76:d4:27:9b:
                    79:52:0c:b0:b3:8c:7d:65:ad:90:87:bd:33:b5:ed:
                    6b:8f:a3:1a:c9:61:ee:cf:ca:dc:8e:74:5b:0f:24:
                    68:62:ed:60:e7:4b:b5:fb:e7:1f:6d:c9:fb:e1:c0:
```

```
                    ec:da:a3:76:83:a5:90:b4:42:d5:78:58:76:a3:6a:
                    17:f6:25:76:2d:ce:bc:9a:2f
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 CRL Distribution Points:
                URI:http://crl.verisign.com/Class3InternationalServer.crl

            X509v3 Certificate Policies:
                Policy: 2.16.840.1.113733.1.7.23.3
                  CPS: https://www.verisign.com/rpa

            X509v3 Extended Key Usage:
               Netscape Server Gated Crypto, Microsoft Server Gated Crypto, TLS
Web Server Authentication, TLS Web Client Authentication
            Authority Information Access:
                OCSP - URI:http://ocsp.verisign.com

-----BEGIN CERTIFICATE-----
MIIEIjCCA4ugAwIBAgIQY66jNJ0mTjBErzo8aNASwzANBgkqhkiG9w0BAQQFADCB
ujEfMB0GA1UEChMWVmVyaVNpZ24gVHJ1c3QgTmV0d29yazEXMBUGA1UECxMOVmVy
aVNpZ24sIEluYy4xMzAxBgNVBAsTKlZlcmlTaWduIEludGVybmF0aW9uYWwgU2Vy
dmVyIENBIC0gQ2xhc3MgMzFJMEcGA1UECxNAd3d3LnZlcmlzaWduLmNvbS9DUFMg
SW5jb3JwLmJ5IFJlZi4gTElBQklMSVRZIExURC4oYyk5NyBWZXJpU2lnbjAeFw0w
MzA4MzAwMDAwMDBaFw0wNDA4MjkyMzU5NTlaMIHJMQswCQYDVQQGEwJCUjEZMBcG
A1UECBMQRGlzdHJpdG8gRmVkZXJhbDERMA8GA1UEBxQIQnJhc2lsaWExHTAbBgNV
BAoUFEJhbmNvIGRvIEJyYXNpbCBTLkEuMRQwEgYDVQQLFAtESVRFQy1HRVRFQzEz
MDEGA1UECxQqVGVybXMgb2YgdXNlIGF0IHd3dy52ZXJpc2lnbi5jb20vcnBhICho
KTAwMSIwIAYDVQQDFBljZXJ0LmJhbmNvZG9icmFzaWwuY29tLmJyMIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQDQ1qXhifja+8pxAwzIPwgsFRO1Wac7BAq8ehir
PUJ+UqSbq64+FjK2w7fTA64lAuwtxafl47ondtQnm3lSDLCzjH1lrZCHvTO17WuP
oxrJYe7PytyOdFsPJGhi7WDnS7X75x9tyfvhwOzao3aDpZC0QtV4WHajahf2JXYt
zryaLwIDAQABo4IBFjCCARIwCQYDVR0TBAIwADALBgNVHQ8EBAMCBaAwRgYDVR0f
BD8wPTA7oDmgN4Y1aHR0cDovL2NybC52ZXJpc2lnbi5jb20vQ2xhc3MzSW50ZXJu
YXRpb25hbFNlcnZlci5jcmwwRAYDVR0gBD0wOzA5BgtghkgBhvhFAQcXAzAqMCgG
CCsGAQUFBwIBFhxodHRwczovL3d3dy52ZXJpc2lnbi5jb20vcnBhMDQGA1UdJQQt
MCsGCWCGSAGG+EIEAQYKKwYBBAGCNwoDAwYIKwYBBQUHAwEGCCsGAQUFBwMCMDQG
CCsGAQUFBwEBBCgwJjAkBggrBgEFBQcwAYYYaHR0cDovL29jc3AudmVyaXNpZ24u
Y29tMA0GCSqGSIb3DQEBBAUAA4GBAIKkLDQ+I14XJKJ8qlDdp158Ue5+0u12B3Mu
lKizewMRNv/Kt/VuGgupeHQD8ZeAXyvnppmYuu3gVT69zkA897GDIWt39NiPnoBt
mGJM4lrSe4CV6I1dqG7LenRTVM0uJYSFWuGP8ZqMafJxU0RyELT0BabJk+w5+BDS
pWM5MTBx
-----END CERTIFICATE-----
Torsec Root CA
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
       Issuer: C=IT, ST= , L= , O=Polito, OU=DAUIN, CN=Certification Authority/
emailAddress=marius@passito.polito.it
        Validity
            Not Before: Sep 13 09:06:25 2001 GMT
```

```
            Not After : Sep 13 09:06:25 2002 GMT
        Subject: C=IT, ST= , L= , O=Polito, OU=DAUIN, CN=Certification Authority/
emailAddress=marius@passito.polito.it
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c3:2d:1e:b9:e8:b3:fa:43:ae:9e:9a:57:2e:e6:
                    18:f7:7c:64:8a:38:64:88:19:23:74:c8:02:8d:8f:
                    c9:27:c6:c8:35:97:fd:70:d1:3b:2a:56:19:b3:59:
                    65:bb:59:93:73:07:a1:11:26:35:21:4d:67:6f:e7:
                    2c:28:b6:4e:dc:19:1c:a0:79:f2:d0:2b:3d:1e:e6:
                    47:9c:3e:27:fa:5b:00:39:f8:de:0f:62:c8:e0:7b:
                    5a:3b:ca:e4:4f:8b:74:9c:58:7a:4a:3e:f9:5a:a5:
                    ed:1e:9a:da:61:72:d3:e0:4d:49:81:53:66:b7:62:
                    06:c8:50:8b:75:8e:55:bb:d1
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                16:00:CD:5D:45:8C:A2:A4:FD:0D:21:36:8E:90:DC:C1:51:21:A6:52
            X509v3 Authority Key Identifier:
              keyid:16:00:CD:5D:45:8C:A2:A4:FD:0D:21:36:8E:90:DC:C1:51:21:A6:52
               DirName:/C=IT/ST= /L= /O=Polito/OU=DAUIN/CN=Certification Autho-
rity/emailAddress=marius@passito.polito.it
                serial:00

            X509v3 Basic Constraints:
                CA:TRUE
```

III.1.2 Tumbleweed (web)

Este certificado foi obtido diretamente do site da empresa Tumbleweed no endereço www.tumbleweed.com. O foco desta empresa é certificação digital e ela opera como uma AR para a Verisign. O conteúdo do certificado é apresentado a seguir.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            13:b4:58:3a:fe:01:71:f8:c5:ff:51:76:61:a6:dc:20
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=RSA Data Security, Inc., OU=Secure Server Certification
Authority
        Validity
            Not Before: Aug 22 00:00:00 2003 GMT
            Not After : Sep 12 23:59:59 2005 GMT
      Subject: C=US, ST=California, L=Redwood City, O=Tumbleweed Communications
Corp., OU=ITOPS, CN=www.tumbleweed.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
```

```
                    00:ab:a1:75:08:72:e5:1e:58:0e:db:6c:22:c3:45:
                    55:9e:19:05:e9:7c:b6:8f:49:22:4c:f4:86:cb:d0:
                    c5:22:f4:88:09:4d:ad:03:6d:17:5f:71:6c:5e:ad:
                    75:e7:14:13:3a:06:5f:8e:e9:6f:48:b5:16:99:45:
                    98:d4:84:24:47:ca:43:23:e5:02:92:85:0f:bc:3a:
                    72:bf:59:fe:6a:e9:ee:d7:8c:cc:3c:d9:18:b1:40:
                    da:53:45:8d:d3:c3:15:c9:cc:ad:b0:e8:46:cb:61:
                    b4:9a:67:bd:80:ba:4e:22:45:be:9f:0c:8f:d0:7d:
                    1f:15:31:82:4d:5a:7f:46:4d
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 CRL Distribution Points:
                URI:http://crl.verisign.com/RSASecureServer.crl

            X509v3 Certificate Policies:
                Policy: 2.16.840.1.113733.1.7.23.3
                  CPS: https://www.verisign.com/rpa

            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            2.16.840.1.113733.1.6.15:
                ..867531600
            Authority Information Access:
                OCSP - URI:http://ocsp.verisign.com

-----BEGIN CERTIFICATE-----
MIIDgzCCAvCgAwIBAgIQE7RYOv4BcfjF/1F2YabcIDANBgkqhkiG9w0BAQUFADBf
MQswCQYDVQQGEwJVUzEgMB4GA1UEChMXUlNBIERhdGEgU2VjdXJpdHksIEluYy4x
LjAsBgNVBAsTJVNlY3VyZSBTZXJ2ZXIgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkw
HhcNMDMwODIyMDAwMDAwWhcNMDUwOTEyMjM1OTU5WjCBkDELMAkGA1UEBhMCVVMx
EzARBgNVBAgTCkNhbGlmb3JuaWExFTATBgNVBAcUDFJlZHdvb2QgQ2l0eTEoMCYG
A1UEChQfVHVtYmxld2VlZCBDb21tdW5pY2F0aW9ucyBDb3JwLjEOMAwGA1UECxQF
SVRPUFMxGzAZBgNVBAMUEnd3dy50dW1ibGV3ZWVkLmNvbTCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEAq6F1CHLlHlgO22wiw0VVnhkF6Xy2j0kiTPSGy9DFIvSI
CU2tA20XX3FsXq115xQTOgZfjulvSLUWmUWY1IQkR8pDI+UCkoUPvDpyv1n+aunu
14zMPNkYsUDaU0WN08MVycytsOhGy2G0mme9gLpOIkW+nwyP0H0fFTGCTVp/Rk0C
AwEAAaOCARAwggEMMAkGA1UdEwQCMAAwCwYDVR0PBAQDAgWgMDwGA1UdHwQ1MDMw
MaAvoC2GK2h0dHA6Ly9jcmwudmVyaXNpZ24uY29tL1JTQVNlY3VyZVNlcnZlci5j
cmwwRAYDVR0gBD0wOzA5BgtghkgBhvhFAQcXAzAqMCgGCCsGAQUFBwIBFhxodHRw
czovL3d3dy52ZXJpc2lnbi5jb20vcnBhMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggr
BgEFBQcDAjAZBgpghkgBhvhFAQYPBAsWCTg2NzUzMTYwMDA0BggrBgEFBQcBAQQo
MCYwJAYIKwYBBQUHMAGGGGh0dHA6Ly9vY3NwLnZlcmlzaWduLmNvbTANBgkqhkiG
9w0BAQUFAAN+AGpxSlgzyJIGRdHiSgcrX3pwmM5jLDaROqMFsrLOrm2bxwo9I9+m
i4W0Z4jnFg4HmHfHDSlrZq/xZDQWn+v57savoz3wTBhD6qNYrYCe5Tbc6ijrxZzt
iQNa9CeTmd5RC1qWgMXcj3CW1osXwQe0Xm5XLaAJ7gyNoY0h9NJE
-----END CERTIFICATE-----
```

III.1.3 Verisign Class 1 CA

Este certificado é dos muitos da Verisign existentes no Windows 2000. Ele foi obtido exportado com o auxílio do Internet Explorer. O conteúdo do certificado é apresentado a seguir.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            0d:8b:4f:ee:aa:d2:18:5b:f4:75:6a:9d:29:e1:7f:fb
        Signature Algorithm: md2WithRSAEncryption
       Issuer: C=US, O=VeriSign, Inc., OU=Class 1 Public Primary Certification
Authority
        Validity
            Not Before: May 12 00:00:00 1998 GMT
            Not After : May 12 23:59:59 2008 GMT
      Subject: O=VeriSign, Inc., OU=VeriSign Trust Network, OU=www.verisign.com/
repository/RPA Incorp. By Ref.,LIAB.LTD(c)98, CN=VeriSign Class 1 CA Individual
Subscriber-Persona Not Validated
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:bb:5a:44:8a:04:16:bb:55:fd:03:7a:8a:2d:94:
                    4f:15:78:36:b8:0d:4a:b2:6f:9c:54:bf:bc:e8:77:
                    2a:9d:b9:f0:68:bb:95:d9:31:41:70:7a:81:4b:b9:
                    48:13:56:2d:c7:08:e1:84:42:ab:c0:a2:92:ab:44:
                    5c:aa:42:f0:82:0e:02:e9:2f:fb:c2:3b:bb:be:c9:
                    27:0a:5d:b6:b0:36:42:33:b5:6e:54:88:4f:87:4a:
                    bf:19:da:f9:15:e8:0f:87:b6:1c:e3:cc:c6:9a:8e:
                    7f:6a:24:92:e3:fc:e0:65:ba:a7:b1:7e:ef:c9:db:
                    37:6a:c8:4a:c8:09:06:e4:99
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Netscape Cert Type:
                SSL CA, S/MIME CA
            X509v3 CRL Distribution Points:
                URI:http://crl.verisign.com/pca1.1.1.crl

            X509v3 Certificate Policies:
                Policy: 2.16.840.1.113733.1.7.1.1
                  CPS: www.verisign.com/repository/RPA

            X509v3 Basic Constraints:
                CA:TRUE, pathlen:0
            X509v3 Key Usage:
                Certificate Sign, CRL Sign
```

III.2 Scripts Utilizados

O programa make-certs.bat (seção C.2.1) foi desenvolvido e utilizado para gerar em lote os certificados utilizados no processo de simulação do tempo de revogação e do tamanho das listas resultante. A localização onde os certificados serão salvos é definida dentro do programa e a linha de comando a ser usada é a seguinte:

`make-certs <num-inicial> <num-final>` onde:

`num-inicial` é o número de série do primeiro certificado a ser gerado,

`num-final` é o número de série do último certificado a ser gerado.

Para o procedimento de obtenção e processamento das LCRs foram desenvolvidos dois scripts, o primeiro, em C shell chamado down_crl.sh é usado para obter as listas dos seus locais de origem e organiza-las numa estrutura de diretório para que possam ser processados pelo segundo programa o gen-datfile.php (seção C.2.2), escrito em PHP que percorre cada entrada e gera o arquivo de dados para que o GNU Plot possa gerar os gráficos. O script do GNU Plot chamado gen-images.plt é apresentado na seção C.2.3. O down_crl.sh é ativado periodicamente por uma entrada no crontab.

III.2.1 make-certs.bat

```
@echo OFF
REM script para gerar certificados em lote
REM parametros de entrada:
REM %1 = numero inicial da sequencia a ser gerada
REM %2 = numero final da sequencia a ser gerada

set CERTPATH=.\_crls
set PROGPATH=..\_progs\openssl-0.9.7c\

REM ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
echo "Gerando Certificados"

FOR /L %%i IN (%1,001,%2) DO (

echo "==> Gerando certificado %%i"

echo "==> Gerando para de chaves RSA"
%PROGPATH%\openssl genrsa -des3 -passout pass:12345 -out %CERTPATH%\%%i.key.pem
1024
echo "==> OK"

echo "==> Criando as requisições dos certificados"
```

```
%PROGPATH%\openssl req -batch -subj /commonName=1234567890/countryName=br/sta-
teOrProvinceName=RiodeJaneiro/emailAddress=1234567890/organizationName=COPPE\/
UFRJ -new -key %CERTPATH%\%%i.key.pem -out %CERTPATH%\%%i.req.pem -extensions
server_ext -config %CERTPATH%\_openssl-spec.txt
echo "==> OK"

echo "==> Assinando os certificados"
%PROGPATH%\openssl ca -batch -passin pass:12345 -in %CERTPATH%\%%i.req.pem -out
%CERTPATH%\%%i.pem -extensions server_ext -config %CERTPATH%\_openssl-spec.txt
echo "==> OK"

)
echo "Terminado."
```

III.2.2 down_crl.sh

```
#!/bin/sh
# shell script para obter e organizar as LCRs obtidas
# por Alessandro Martins - martins@ufrj.br - marco/2004

MYPATH='/extra1/user/martins/download'
DATE=`date +%Y%m%d-%H%M%S`

# Processa as LCRs da ACNAME
ACNAME=verisign
NAME=$DATE.tar.gz
if [ ! -d $MYPATH/$ACNAME ] ; then
   mkdir -p $MYPATH/$ACNAME
fi
wget -mr -np crl.$ACNAME.com
mv crl.$ACNAME.com $MYPATH/$ACNAME/$NAME

# Processa as LCRs da ACNAME
ACNAME=rsasecurity
NAME=$DATE.tar.gz
if [ ! -d $MYPATH/$ACNAME ] ; then
   mkdir -p $MYPATH/$ACNAME
fi
wget -mr -np crl.$ACNAME.com
mv crl.$ACNAME.com $MYPATH/$ACNAME/$NAME

# Processa as LCRs da ACNAME
ACNAME=unicert
NAME=$DATE.tar.gz
if [ ! -d $MYPATH/$ACNAME ] ; then
        mkdir -p $MYPATH/$ACNAME
fi
wget -mr -np www.unicert.com.br/crl/2590C.crl
wget -mr -np www.unicert.com.br/crl/2590A.crl
mv www.$ACNAME.com.br $MYPATH/$ACNAME/$NAME

# FIM
```

III.2.3 gen-datfile.php

```
<?PHP
// script para processar as LCRs obtidas pelo down_crl.sh
// por Alessandro Martins - martins@ufrj.br - marco/2004

// flag de DEBUG
$debug = 0;

$result_dir = "_results\\"; // onde os resultados serao colocados
$source_dir = ".\\";// onde os diretorio dos downloads estao
$crl_url = "crl.verisign.com";  // subdir onde estao as CRLs

// remove os logs anteriores
echo "=> Removendo os logs anteriores em $result_dir";
$result = get_file_list($result_dir);
foreach ($result as $file) {

   $extension = substr($file,sizeof($file)-6,5);
   if ($extension == ".dat1") {
      unlink($result_dir.$file);
   }
}
echo "\t\t Feito. \n";

// Gera os novos logs
echo "=> Lendo o dir $source_dir e gerando logs, aguarde. \n";
$result = get_file_list($source_dir);

//ordena a lista
sort ($result,SORT_NUMERIC);

// para cada data de download, faca
foreach ( $result as $download_date) {

   // verifica se eh um diretorio
   if (is_dir($download_date) && ($download_date."\\" != $result_dir)) {

      if ($debug)  echo "\n=> Processando $download_date \n";

      $place_of_crls = $source_dir.$download_date."\\".$crl_url."\\";
      $list_of_crls = get_file_list($place_of_crls);

      // para cada crl encontrada, crie um aquivo com o seu nome
      // e dentro dele armazene a data e o tamanho do aquivo no formato
      // mmdd tamanho
      foreach ( $list_of_crls as $crl_name )
      // trata apenas as CRL, outros arquivos possuem terminacao diferente de .crl
      if (substr($crl_name,sizeof($crl_name)-5,4) == ".crl") {
         // nome do arquivo de saida
         $filename = substr($crl_name,0,sizeof($crl_name)-5);
         $fp = @fopen($result_dir.$filename.".dat1", "a");

         // formato da data de saida
         $date_formated = substr($download_date,4,4);
```

```
        $line_formated = $date_formated."\t".filesi-
ze($place_of_crls."\\".$crl_name)."\n";
        if ($debug)
           echo $crl_name."\t".$line_formated;
        else
           echo ".";

        fwrite($fp,$line_formated);
        fclose($fp);
     }

     if ($debug) echo "=> Feito. \n";
   }

}; // fim do procedimento
echo "\n=> Processamento terminado com sucesso. \n";


// funcao auxiliar para obter a lista de entradas num diretorio.
function get_file_list ($dir_name) {

   $files_in_dir = array();

   // o @ inibe a msg de erro, mas o proximo if pega o erro
   $handle = @opendir($dir_name);
   if (!$handle) {
      echo "Ocorreu um erro ao executar o processo\n"; exit;
   }
   // debug
   if ($debug) echo "Processando o diretorio $dir_name \n";

   while (false !== ($file = readdir($handle))) {

      if ($file != "." && $file != "..") {

         // pega o nome do arquivo sem a extensao
         //$files_in_dir[] = substr($file,0,sizeof($file)-5);

         // arquivo com extensao ou diretorio
         $files_in_dir[] = $file;

      } // end if
   } // end while

   closedir ($handle);

   return $files_in_dir;

} // end get_file_list

// atencao nao pode haver linha em branco ao final deste arquivo
?>
```

III.2.4 Gen-images.plt

```
#
# Script para gerar os graficos dos parametros das LCRs obtidas
# usando o gnuplot com o comando pgnuplot <nome-do-arquivo>
# por Alessandro Martins - martins@ufrj.br - marco/2004
#

reset

# configuracao para salvar as imagens
set terminal push
set terminal gif medium size 820,400

# formato dos arquivos *.dat1 de entrada
# data tamanho-do-arquivo

# primeiro grafico
set output '_constantes.gif'
set autoscale
set title "Tamanho da LCR no tempo"
set ylabel 'Tamanho da CRL (bytes)'
set yrange [320:1700]
set xlabel 'Data (formato dd/mm)'
set xdata time
set timefmt "%m%d"
set format x "%d/%m"
plot 'ThawteTimestampingCA.dat1' using 1:2,\
'ThawtePremiumServerCA.dat1' using 1:2,\
'WAPSecureServer.dat1' using 1:2
set output

# segundo grafico
set output '_degrau.gif'
set autoscale
set title "Tamanho da LCR no tempo"
set xlabel 'Data (formato dd/mm)'
set xdata time
set timefmt "%m%d"
set format x "%d/%m"
set ylabel 'Tamanho da CRL (bytes)'
plot 'Class3IntranetServer.dat1' using 1:2 lt 3 pt 6
set output

# terceiro grafico
set output '_crescente.gif'
set autoscale
set title "Tamanho da LCR no tempo"
set xlabel 'Data (formato dd/mm)'
set xdata time
set timefmt "%m%d"
set format x "%d/%m"
set ylabel 'Tamanho da CRL (bytes)'
plot  'Class3CodeSigning2001.dat1' using 1:2,\
'ThawtePersonalFreemailIssuingCA.dat1' using 1:2
```

```
set output

# fim
print "\n Terminado."
```

III.3 Programas Desenvoldidos

Para implementar o cliente e o servidor OCSP foram criados dois programas. O código fonte de cada um destes programas é fornecido no CDROM em anexo a esta dissertação e o procedimento de compilação e utilização utilizando cada uma das bibliotecas selecionadas é apresentado nas seções que seguem.

Para gerar os programas utilizando a cryptlib é necessário primeiro compilar a própria biblioteca. Para isto é suficiente seguir os procedimentos específicos para cada plataforma existente no arquivo INSTALL existente na distribuição da biblioteca. O mesmo precisa ser feito para o OpenSSL, caso não seja utilizada a versão já compilada existente no site do projeto.

Após esta fase utiliza-se o makefile desenvolvido para auxiliar na compilação. Os makefile apresentados nas seções correspondentes são específicos para o compilador Microsoft Visual C++ 6.0. Para utiliza-los basta invocar o comando `nmake -f <arquivo.mak>`.

III.3.1 Cliente OCSP usando Cryptlib

A ativação do cliente se da invocando o programa com a seguinte linha:

- Requisições sem assinatura:

```
ocsp_client serverURL caFile eeFile
```

- Requisições assinadas:

```
ocsp_client serverURL caFile eeFile eeKey
```

Os parametros são:

`serverURL` é o endereço do servidor, exemplo localhost:334

`caFile` e `eeFile` são os arquivos contendo respectivamente o certificado da AC e o certificado a ser checado

`eeKey` é o arquivo contendo a chave privada a ser utilizada na assinatura da requisição.

III.3.2 Servidor OCSP usando Cryptlib

A ativação do servidor se da invocando o programa com a seguinte linha:

```
ocsp_server port privKeyFile privKeyPass caFile ocspFile
```

Os parametros são:

`port` porta na qual o servidor irá ouvir as requisições

`privKeyFile` arquivo com a chave privada da AC

`privKeyPass` senha de acesso ao arquivo com a chave privada

`caFile` e `ocspFile` são os arquivos contendo respectivamente o certificado da AC e o certificado do Responder caso não sejam os mesmos.

III.3.3 Cliente OCSP usando OpenSSL

A sintaxe de uso dos programas desenvolvidos com o OpenSSL é a seguinte:

- Requisições sem assinatura:

```
ocsp-c -host serverURL -VAfile caFile -cert eeFile
```

- Requisições assinadas:

```
ocsp_client -host serverURL -VAfile caFile -cert eeFile
-signKey eeKey
```

Os parametros são:

`serverURL` é o endereço do servidor, exemplo localhost:334

`caFile` e `eeFile` são os arquivos contendo respectivamente o certificado da AC e o certificado a ser checado

`eeKey` é o arquivo contendo a chave privada a ser utilizada na assinatura da requisição.

III.3.4 Servidor OCSP usando OpenSSL

A ativação do servidor se da invocando o programa com a seguinte linha:

`ocsp-s -port port -index file -CAfile caFile -VAfile ocs-pFile`

Os parametros são:

`port` porta na qual o servidor irá ouvir as requisições

`index` arquivo com as informações sobre revogação

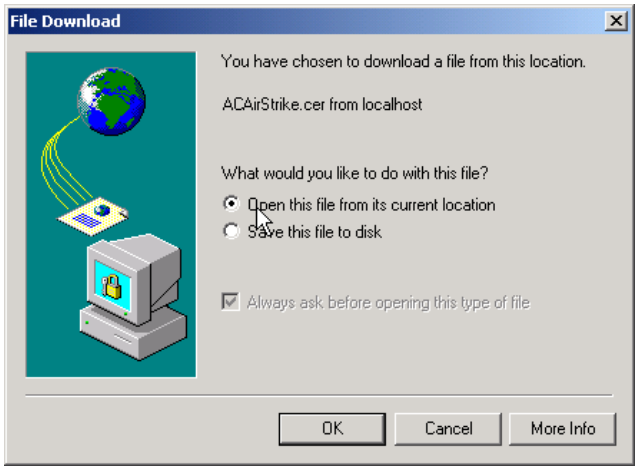
`caFile` e `ocspFile` são os arquivos contendo respectivamente o certificado da AC e o certificado do Responder caso não sejam os mesmos.

Apêndice IV: Instalação de Certificados

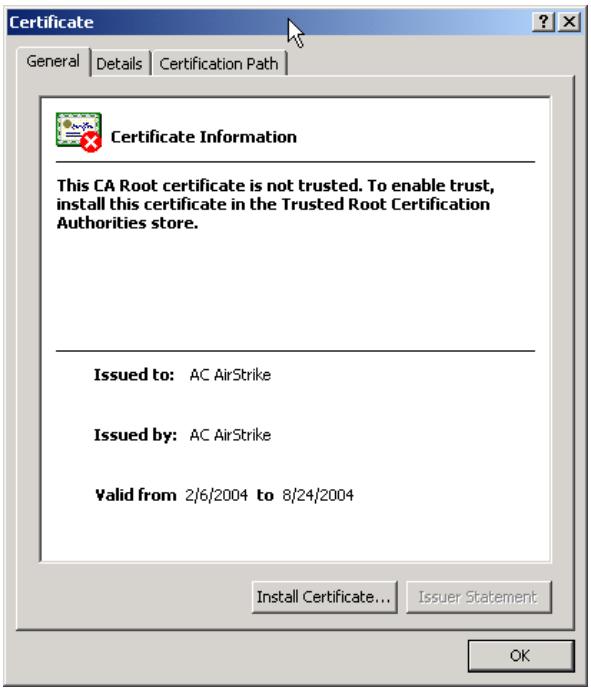
Este apêndice descreve os procedimentos necessários para importar os certificados das autoridades e dos usuários para os cliente web mais comuns: o Internet Explorer e o Netscape.

IV.1 Instalação dos Certificado no IE

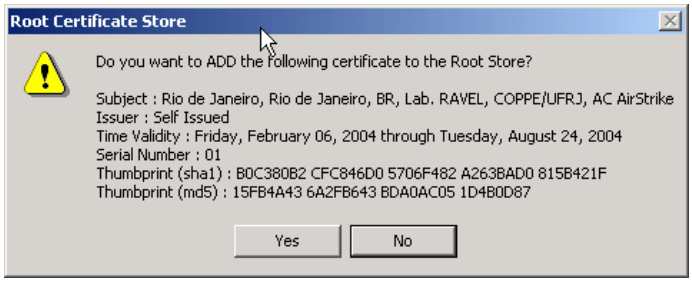
Primeiro é preciso obter o certificado da Autoridade Certificadora Raiz. Isso pode ser feito usando a opção (7) do menu lateral da página de acesso a interface web da ICP. Na janela apresentada, selecione abrir o arquivo, como mostrado na figura que segue..



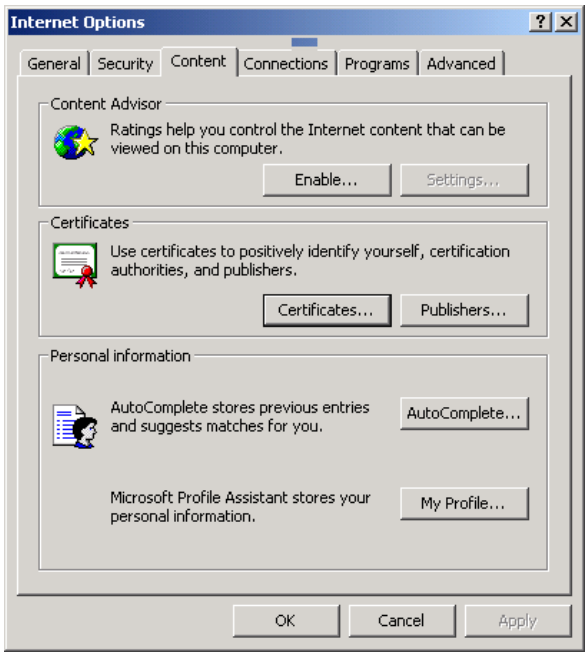
Em seguida, na barra inferior da janela que surge existe a opção de Instalar o Certificado.



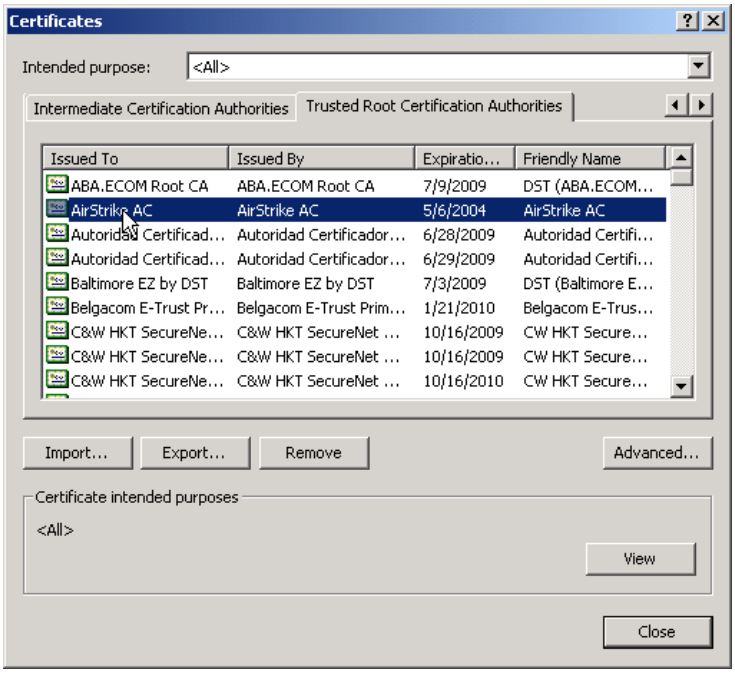
O processo que segue termina com a janela seguinte, solicitando a confirmação final do processo.



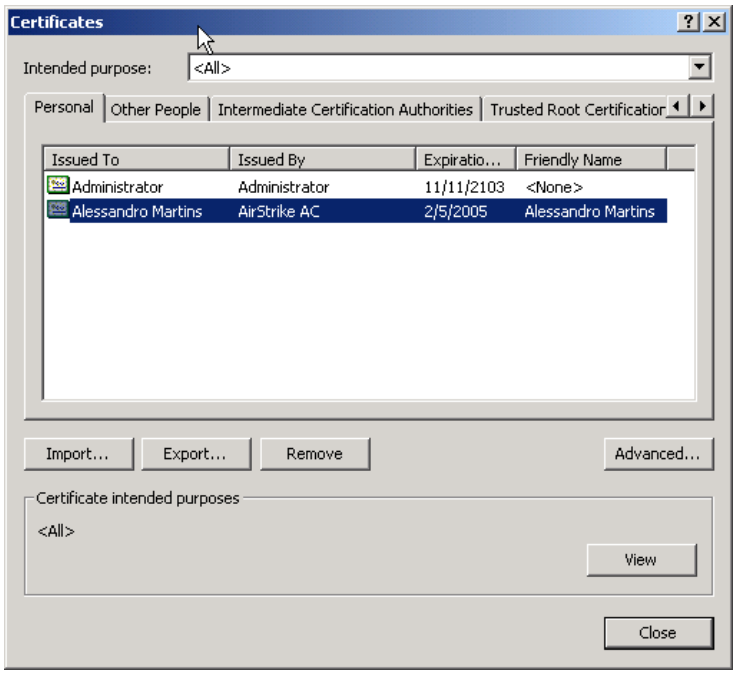
Para confirmar a correta instalação basta navegar pelos menus `Tools -> Internet Options` até a janela da figura abaixo.



Utilizando a opção `Certificates` é mostrada a janela que segue e na última aba o nome da AC instalada deverá existir.



O mesmo procedimento deve ser seguido para que seja feita a instalação do certificado do cliente, finalizando na janela que segue.



IV.2 Instalando Certificados no Netscape 7+ ou Mozila 1+

Um procedimento semelhante pode ser seguido para executar a instalação dos certificados nestes clientes. Como o Mozila 1.0 e o Netscape 7+ possuem suporte nativo para o protocolo OCSP, torna-se necessário para ativar esta funcionalidade instalar também o certificado do OCSP *Responder*, que pode ser obtido pela opção (1) do menu principal da interface web. Nenhum dos produtos da Microsoft, em especial o Internet Explorer, possue suporte nativo ao protocolo OCSP, por isso este passo não foi mostrado na seção correspondente.

Para executar a importação, siga até a interface de certificados (vide Figura IV-1) caminhando pelas opções `Edit -> Preferences -> Privacy & Security -> Certificates -> Manage Certificates`.

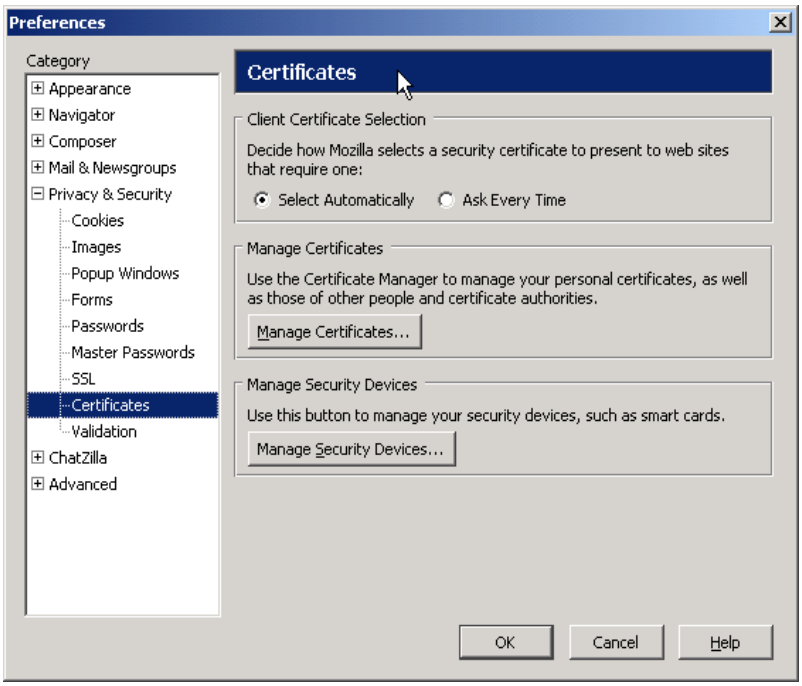


Figura IV-1. Interface de gerenciamento de certificados do Mozila

A opção `Import` na barra da janela apresenta um menu para a seleção do arquivo e termina com uma janela de confirmação, vista na figura seguinte.

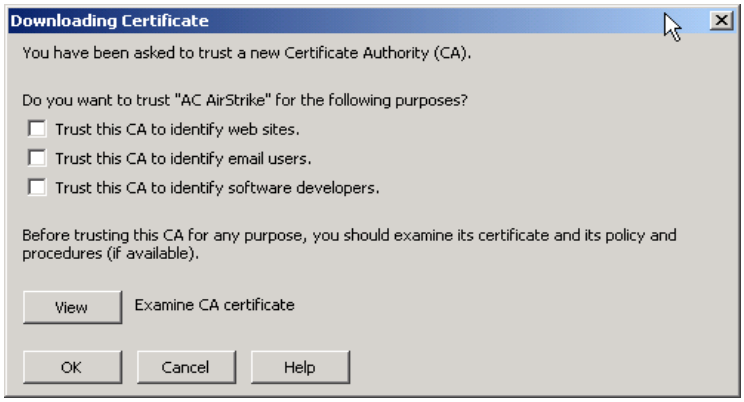
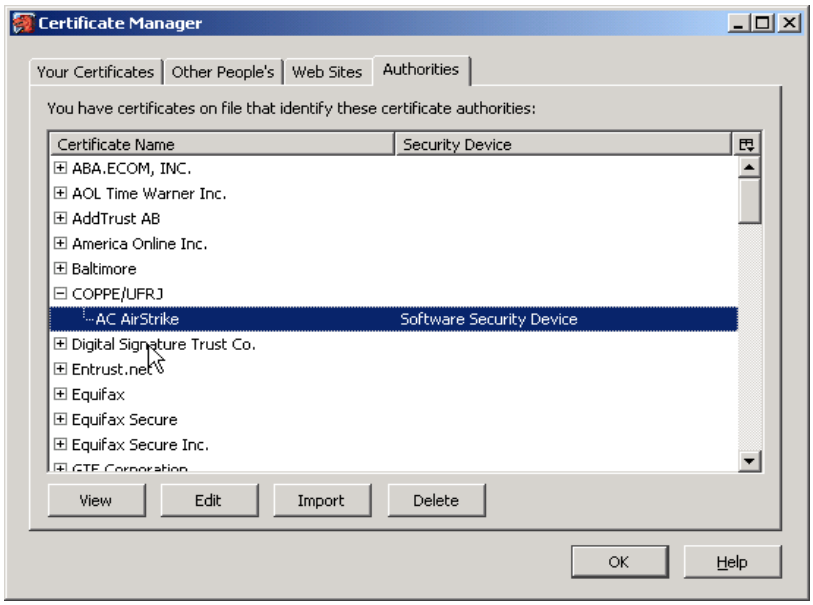


Figura IV-2. Interface de importação do Mozila

A figura que segue mostra o resutado da importação e também pode ser usada para verificar se um certificado já esta instalado.



O mesmo processo deve ser executado para importar o certificado do OCSP *Responder* e do próprio cliente.

A figura abaixo mostra a configuração das opções de verificação de certificados.

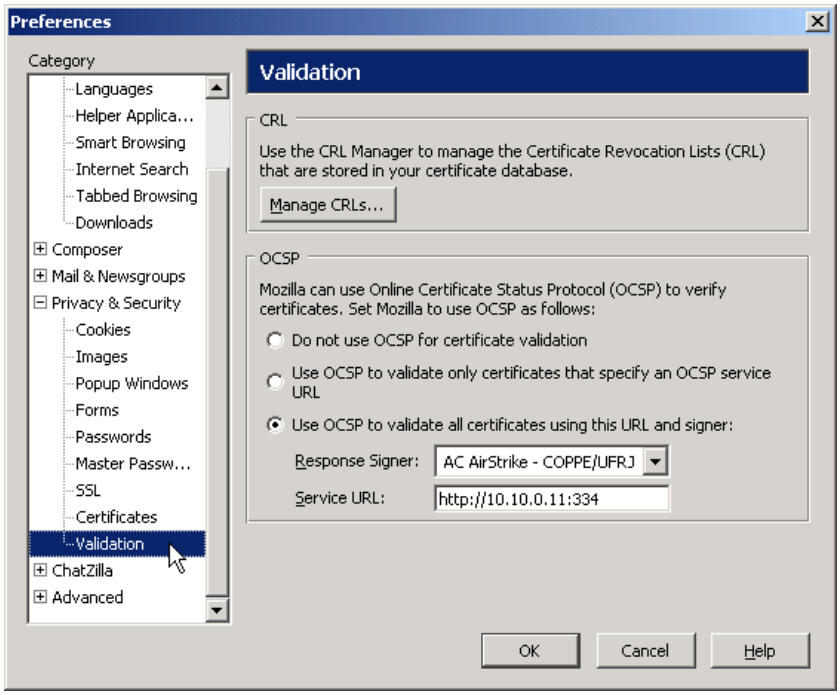


Figura IV-3. Configuração de verificação de certificados do Mozila

Nela pode-se selecionar qual dos métodos de verificação será utilizado. Note que as opções são mutuamente exclusivas, ou seja, não é possível ativar mais de uma opção.

Apêndice V: Instalação da NewPKI

Este apêndice descreve os procedimentos necessários para instalar e configurar a NewPKI.

V.1 Instalação e Configuração

A instalação da solução NewPKI exige que o ambiente já tenha instalado e devidamente configurado um servidor Apache 1.3.23 com suporte a PHP 4 ou mais atual e o banco de dados MySQL numa versão superior a 3.23.49.

O processo de instalação do servidor é descrito na documentação distribuida com a aplicação. Já instalação da interface WEB é feita colocando os scripts PHP num diretório apropriado da árvore de diretórios do Apache. A configuração dessa interface é feita alterando os campos que definem o endereço do servidor do banco de dados, a conta e a senha de acesso. Isso é feito alterando o arquivo conf.inc.php na raiz da árvore de diretório gerada após a expansão do arquivo contendo os scripts.

O processo de instalação e configuração do módulo servidor da NewPKI pode ser feito com base no arquivo README presente na distribuição. No ambiente Windows existe um programa de instalação que executa todo o processo e do mesmo modo, para a plataforma Linux, existe um pacote pronto para a instalação. Caso haja a necessidade é possível, utilizando um outro formato da distribuição, compilar o servidor seguindo os passos normais desse processo num ambiente Unix: primeiro executa-se um processo de identificação do sistema, pelo comando `Configure`, em seguida compila-se os fontes com o comando `make` e finalmente, executa-se a instalação com o comando `make install`.

Após a instalação ter sido realizada o arquivo de configuração do servidor (chamado config.conf) precisa ser alterado para refletir as mesmas configurações feitas para a interface WEB, ou seja, é necessário definir o endereço, o login e a senha de acesso ao banco de dados.

V.2 Interface do Sistema

A solução NewPKI é composta pode 3 módulos, com as seguintes funções:

• Interface web: canal usado para a interação do usuário com o sistema. Através dela o usuário pode obter os certificados das autoridadees existentes, solicitar e acompanhar o estágio do processo de emissão do seu certificado. Esta interface é descrita em detalhes na seção seguinte.

• Servidor: Executa toda a interação com o repositório e as ações enviadas pelos 3 atores atravéz dos clientes.

• Cliente: Abriga consigo três interfaces distintas:

- a primeira ativada apenas para os administradores do sistema, responsáveis por estabelecer as configurações, definir as mensagens a serem enviadas e estabelecer as políticas de certificação.
- a segunda para os RA Operators, que entre outras funções, aprovam as requisições feitas pelos cliente via interface web.
- a terceira para os *CA Operators* que possuem como função principal, autorizar a emissão dos certificados aprovados pelos *RA Operators*.

Graficamente o processo pode ser representado pela Figura V-1

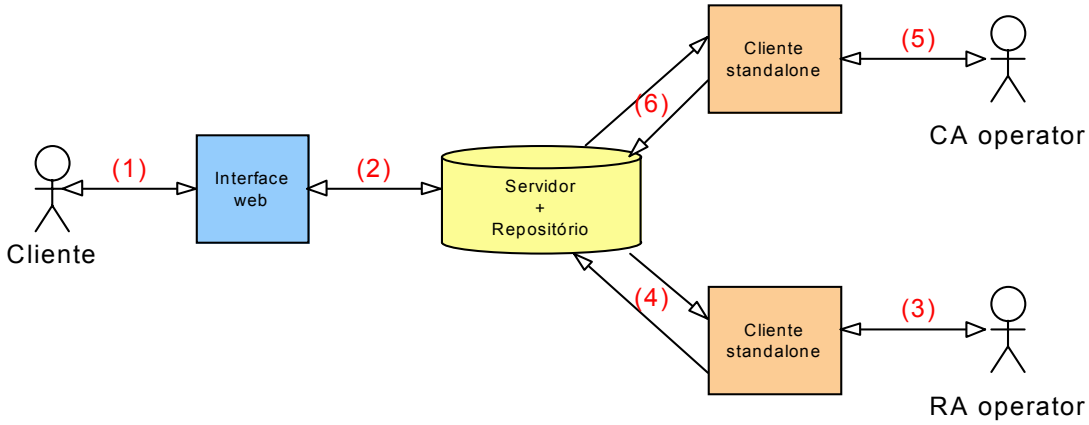


Figura V-1. Diagrama de interação com o NewPKI

Descrição das Mensagens:

(1) - Interação do cliente com a interface web durante os processos de solicitação de certificado e durante a obtenção do status do processo.

(2) - Interação da interface web com o repositório via servidor atendendo as solicitações feitas pelo cliente.

(3) e (4) - Interação do RA Operator com o repositório via servidor na obtenção da relação de solicitações de certificados a serem avaliadas. As decisões tomandas (aceite ou rejeição da requisição) são armazenadas novamente no respositório.

(5) e (6) - Do mesmo modo que o RA Operator, o CA Operator obtem uma lista de requisições aprovadas e decide sobre a emissão ou não dos certificados. O resultado das ações são novamente armazenados no repositório para efeito de informação ao cliente.