Autenticação de Usuários em Ambientes Distribuídos

Maira Trés Medina
Noemi de La Rocque Rodriguez
Departamento de Informática, PUC-Rio
Rua M. São Vicente, 225 - Gávea
22453-900, Rio de Janeiro - RJ
maira,noemi@inf.puc-rio.br

Abstract

Este trabalho discute o problema de distribuição de chaves públicas, usadas para criptografia assimétrica, em ambientes geograficamente distribuídos. A motivação para essa discussão é a provisão de segurança em correio eletrônico. Diversas soluções para o problema são apresentadas, com ênfase especial no uso de certificados emitidos por autoridades de certificação. Aborda-se o problema de validação de certificados entre usuários atendidos por diferentes autoridades de certificação, e descreve-se uma ferramenta que permite a troca segura de chaves nessa situação.

This paper discusses the problem of public key distribution in wide area networks. The goal of this discussion is to better understand the problem in the context of secure electronic mail. Several solutions are presented, with special emphasis on the use of certificates signed by certification authorities. The problem of validating certificates exchanged by users associated to different certification authorities is discussed, and a tool developed for this task is described.

1. Introdução

Nos últimos anos, o correio eletrônico vem cada vez mais se firmando como um mecanismo de comunicação interna em diferentes ambientes empresariais. Isto fez com que a questão de segurança, originalmente desprezada no ambiente Internet, assumisse importância fundamental. Vários sistemas [KPS95] têm sido propostos ou implementados para a provisão de serviços de confidencialidade e autenticação em correio eletrônico. Tipicamente, a provisão destes serviços é baseada em mecanismos de criptografia cuja confiabilidade parece atender as necessidades atuais.

No entanto, o uso de mecanismos de criptografia implica na necessidade de divulgação de chaves entre os parceiros de uma comunicação. O problema da chamada *distribuição de chaves* parece ser no momento, o maior empecilho para o uso generalizado de sistemas de correio eletrônico com segurança.

Este trabalho discute o problema de distribuição de chaves em ambientes geograficamente distribuídos. Para isto, inicialmente é apresentada uma rápida revisão dos mecanismos de criptografia simétrica e assimétrica e de seu uso em comunicação. Na maior parte dos sistemas de correio eletrônico com segurança, a necessidade de distribuição de chaves se refere à chamada chave pública, usada na criptografia assimétrica. Discute-se então os problemas de gerenciamento de chaves públicas e diversas soluções para a sua distribuição, com diferentes níveis de sofisticação. A solução com maior aceitação atual é o uso de certificados emitidos por uma autoridade de certificação. Em um ambiente geograficamente distribuído, a tendência é que seja utilizada não uma, mas um conjunto de autoridades de certificação. Discute-se então como permitir a interação segura entre usuários atendidos por diferentes autoridades de certificação. Finalmente, apresenta-se uma ferramenta desenvolvida com o objetivo de dar suporte a este tipo de interação. Essa ferramenta, desenvolvida em ambiente UNIX e utilizando um serviço de diretórios, constrói uma cadeia de certificados entre duas autoridades arbitrárias, permitindo que um usuário associado a uma destas autoridades valide um certificado emitido pela outra.

2. Sistemas criptográficos simétricos e assimétricos

Criptografia, do grego "*kryptos*=escondido, oculto" e "*grápho*=grafia, escrita" é a arte ou ciência de escrever em cifra ou em código, ou em outras palavras, é um conjunto de técnicas que permite tornar incompreensível um texto originalmente escrito com clareza, a permitir, normalmente, que apenas entidades autorizadas o decifrem e compreendam [Luc85]. Quase sempre a codificação e o deciframento requerem o conhecimento de uma chave, que é uma informação secreta.

O motivo para criptografar textos no contexto de comunicação é a provisão de segurança para a transmissão destes textos sobre canais inseguros. De modo geral, no caso de redes de computadores, a vulnerabilidade do sistema é maior do que em sistemas de computação autônomos, e não se pode resolver o problema de segurança fisicamente. A informação circula em redes públicas de comunicação, o que permite que venha a ser acessada inadequadamente.

A criptografia assimétrica [DH76], baseada em funções matemáticas, promoveu uma revolução, devido ao fato dos algoritmos fazerem uso de duas chaves complementares, ao contrário dos algoritmos simétricos, que utilizam apenas uma chave compartilhada.

A idéia principal do desenvolvimento da criptografia assimétrica foi tentar atacar dois grandes problemas da criptografia simétrica. O primeiro é o da distribuição das chaves, pois sistemas criptográficos simétricos requerem que os dois participantes compartilhem a mesma chave, sendo que um dos dois ou uma terceira entidade deve emití-la. De acordo com Diffie e Hellman (proponentes da criptografia assimétrica), isto nega a verdadeira essência da criptografia [Sta95]: "a habilidade de manter total segredo sobre a comunicação".

O segundo problema que Diffie assinalou foi o da "assinatura digital". Se a criptografia seria amplamente utilizada com propósitos comerciais e privados, então mensagens e documentos eletrônicos necessitariam de um mecanismo equivalente às assinaturas utilizadas em documentos de papel [Sta95].

Segundo [Dif88], duas características tipicas de sistemas criptográficos assimétricos são que qualquer coisa que é criptografada com uma chave só pode ser decriptada com a outra e que é computacionalmente impossível determinar uma chave a partir do conhecimento o algoritmo e da chave complementar.

O uso típico deste par de chaves em comunicação segura é associar a cada usuário ou entidade um par de chaves, sendo que uma destas chaves, chamada *chave privada*, é de conhecimento apenas deste usuário, e a outra, denominada *chave pública*, é divulgada publicamente.

Os serviços básicos fornecidos por sistemas de correio eletrônico com segurança são: confidencialidade, integridade dos dados, autenticação de origem e não repúdio.

Dependendo da aplicação, o emissor de uma mensagem utiliza a sua chave privada, ou a chave pública do receptor, ou ambas, para executar algum tipo de função de criptografia. Nestes termos, o uso de sistemas criptográficos é classificado em três tipos:

1. Confidencialidade: o emissor criptografa a mensagem com a chave pública do receptor.

2. Assinatura digital e integridade: o emissor assina a mensagem com a sua chave privada.

3. Troca de chaves: os dois participantes da comunicação trocam uma chave de seção entre si.

Existem várias condições que são exigidas para que um algoritmo seja caracterizado como algoritmo criptográfico assimétrico. São elas:

1. Deve ser computacionalmente simples gerar o par de chaves (chave pública e chave privada).

2. Deve ser computacionalmente simples para um emissor A, conhecendo a chave pública de B e a mensagem X a ser criptografada, produzir o texto criptografado correspondente: $Y = C_{PbB}(X)$.

3. Deve ser computacionalmente simples para o receptor B decriptar o texto criptografado, usando a sua chave privada, recuperando assim a mensagem original: $X = D_{PrB}(Y) = D_{PrB}[C_{PbB}(X)]$.

4. Deve ser computacionalmente impraticável para um intruso, com o conhecimento da chave pública (PbB), determinar a chave privada (PrB).

5. Deve ser computacionalmente impraticável para um intruso, com o conhecimento da chave pública (PbB) e da mensagem criptografada (Y), recuperar a mensagem original (X).

Convencionou-se reservar o termo "chave secreta" para a chave usada na criptografia simétrica e o termo "chave privada" para a chave mantida só pelo usuário na criptografia assimétrica, sendo que ambas devem permanecer em segredo.

O debate sobre qual método de criptografia é o melhor vem se desenrolando desde que a criptografia assimétrica foi inventada. Não há sentido em responder sobre qual método é o mais seguro. A segurança dos dois métodos consiste no segredo da chave secreta e da chave privada, em criptografia simétrica e assimétrica respectivamente, não do segredo do algoritmo utilizado.

A segurança dos algoritmos é dependente do tamanho da chave a ser usada. A maioria dos algoritmos simétricos trabalha com um tamanho de chave específico, enquanto que algoritmos assimétricos permitem

tamanhos de chaves variáveis. Por exemplo, o algoritmo DES [Sch94] (simétrico) com uma chave de tamanho igual a 56 bits é mais seguro do que o RSA [Sch94] (assimétrico) com uma chave de tamanho igual a 40 bits. Por outro lado, o RSA com uma chave de tamanho igual a 1000 bits será mais seguro que o DES.

Em [Dif88] é destacado um comentário sobre a performance dos algoritmos RSA e DES em relação ao processo de criptografia de um texto. O RSA é 1000 vezes mais lento que o DES, além de tipicamente trabalhar com chaves dez vezes maiores. Sob este aspecto, o custo a ser pago pelas vantagens do RSA é alto. As implementações mais rápidas do RSA criptografam somente alguns milhares de bits por segundo, enquanto que as implementações mais rápidas do DES criptografam vários milhões de bits por segundo. A razão desta disparidade é que o DES foi projetado para ser facilmente implementado em hardware. Esses números incluem a diferença entre o processamento por hardware e software.

A assinatura digital em algoritmos assimétricos, por exemplo, o RSA, permite eficiente mecanismo de autenticação quando utilizada em conjunto com uma função de hash. A função de hash aplicada sobre a mensagem tem por finalidade produzir um código de tamanho fixo, e é esperado que cada mensagem produza um código de hash único. A eficiência é devido ao fato do algoritmo não criptografar a mensagem toda, mas sim, criptografar apenas o código de hash relativo a mensagem. Isto é, a assinatura digital é realizada sobre o código de hash da mensagem.

Os dois tipos de criptografia trabalham bem em conjunto, pois cada um apresenta uma vantagem sobre o outro. Este esquema é ilustrado nas Figuras 2.1 e 2.2. A vantagem da criptografia simétrica está na eficiência da criptografia/decriptação de textos e arquivos. Por outro lado, a vantagem da criptografia assimétrica está na distribuição de chaves e na integridade dos dados.

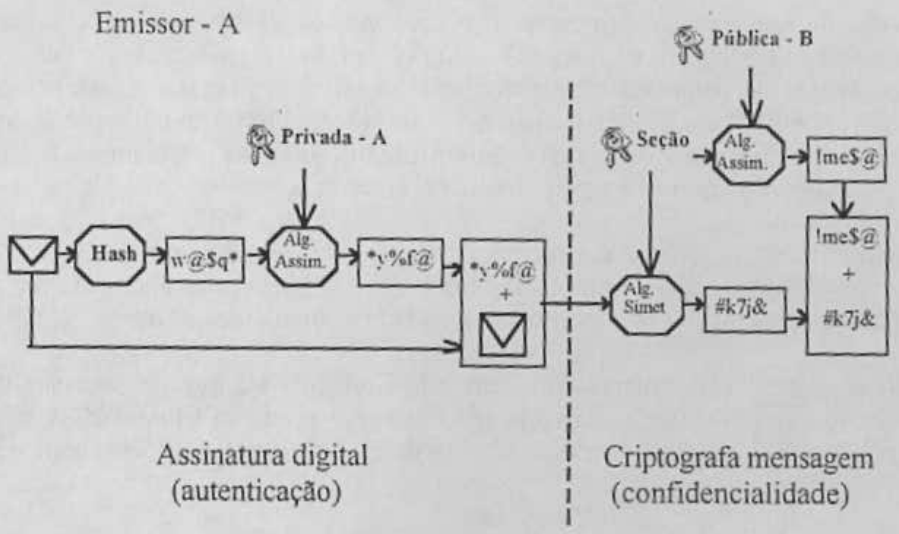

Figura 2.1 - Criptografia assimétrica e simétrica - emissor da mensagem.

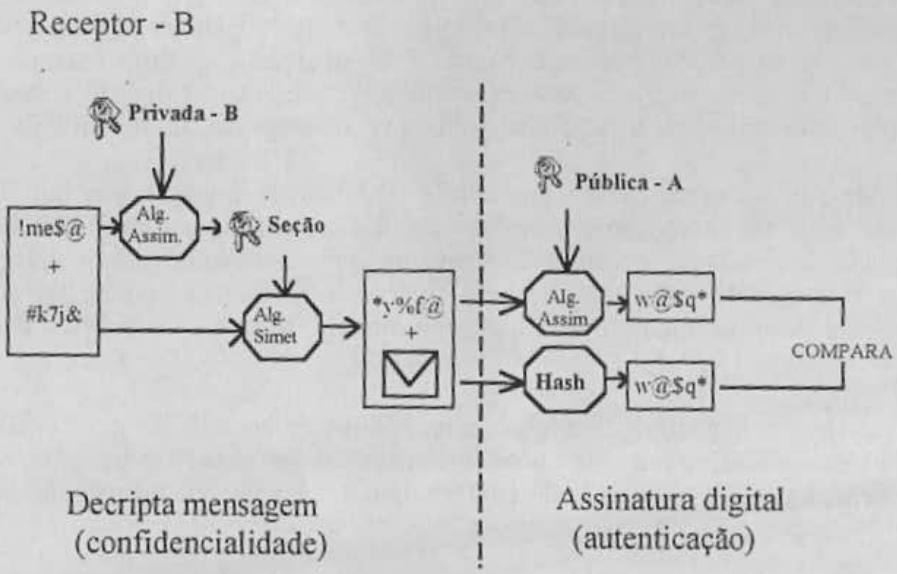

Figura 2.2 - Criptografia assimétrica e simétrica - receptor da mensagem.

No esquema híbrido mostrado nas Figuras 2.1 e 2.2, tem-se que os processos de assinatura digital (integridade dos dados e autenticação) e de troca de chaves são realizados por sistemas assimétricos, e o processo de confidencialidade dos dados fica a cargo dos sistemas simétricos. Na Figura 2.2, o processo de decriptar a mensagem ocorre antes do processo de verificação da assinatura.

3. Gerenciamento de chaves públicas

O único método conhecido para a transmissão segura de dados entre os parceiros de uma comunicação sobre canais inseguros é a utilização da criptografia.
Um dos maiores problemas na utilização de mecanismos de criptografia reside na necessidade de distribuição das chaves a serem usadas. Esta necessidade gera um problema circular. A fim de se obter uma comunicação segura sobre canais inseguros, os usuários devem primeiro trocar informações sobre as chaves públicas a serem usadas na comunicação. Se nenhuma alternativa para canais inseguros existe, então uma troca segura da informação da chave apresenta o mesmo problema de segurança de uma comunicação segura.

O conceito de distribuição pública de chaves secretas foi introduzido em 1976 por Diffie e Hellman, no mesmo artigo em que foi introduzido o conceito de criptografia assimétrica [DH76]. O algoritmo proposto por Diffie e Hellman representa um protocolo, onde o principal objetivo é o estabelecimento de um segredo compartilhado entre duas entidades que desejam se comunicar de modo seguro, isto é, o estabelecimento de uma chave secreta. A segurança do algoritmo Diffie-Hellman é baseado na dificuldade do cálculo de logaritmos discretos e no cálculo de exponenciação utilizando-se uma matemática simples. Este algoritmo pode ser melhor estudado em [DH76, Sch94, Nec92].

O importante é que as entidades da comunicação trocam entre si mensagens que são transmitidas sobre canais inseguros para o estabelecimento do segredo (chave secreta). Outras entidades podem ter conhecimento destas mensagens, pois não serão capazes de computar o segredo. Se o objetivo principal é o estabelecimento de um segredo sobre canais inseguros, então ele é alcançado com este protocolo. Uma desvantagem deste protocolo é que ele não apresenta nenhum suporte para a autenticação entre as entidades, devendo ser utilizado em conjunto com outros protocolos que ofereçam suporte à autenticação.

Em sistemas criptográficos assimétricos, o gerenciamento de chaves está relacionado com a troca da componente pública do par de chaves gerado. Este problema parece ser simples, pois a chave pública não requer nenhum segredo em seu armazenamento ou na sua transmissão sobre canais inseguros.

A idéia é que estas chaves públicas possam estar disponíveis em um diretório, ou que possam ser trocadas diretamente entre os usuários. Sempre que um usuário A deseja se comunicar com um usuário B, A só precisa encontrar a chave pública de B no diretório ou recebê-la diretamente de B, criptografar a mensagem com esta chave e enviá-la para B.

A segurança de sistemas assimétricos é criticamente dependente da seleção correta da chave pública pelo emissor da mensagem. Se a chave pública selecionada é incorreta, a proteção oferecida pela criptografia assimétrica é perdida. Chaves incorretas podem gerar resultados indesejáveis, como por exemplo, se um usuário A forjar um comunicado divulgando sua chave pública como sendo a chave pública de B, A poderá ler todas as mensagens confidenciais para B geradas por esta chave. Esta situação é possível sempre que o esquema de distribuição por difusão é usado, como ocorre, por exemplo, entre os usuários do sistema PGP [Zim93]. Os usuários adotam a prática de adicionar a sua chave pública às mensagens que enviam.

Um maior grau de segurança pode ser alcançado pela manutenção das chaves públicas em um diretório dinâmico disponível publicamente, com a manutenção e a distribuição das chaves neste diretório público sob responsabilidade de alguma entidade de confiança do sistema. Caso um intruso obtenha acesso indevido ao diretório, o intruso novamente será capaz de forjar qualquer participante, e consequentemente ter acesso às mensagens enviadas para este participante. O trabalho do intruso, no entanto, torna-se mais difícil do que no caso da distribuição por difusão.

Para aumentar a segurança do esquema acima, deve-se adotar uma autoridade central. Esta autoridade, que representa uma autoridade de chaves públicas, mantém um diretório dinâmico das chaves públicas de todos os usuários. Cada usuário conhece a chave pública da autoridade central. Um cenário típico é ilustrado na Figura 3.1, que é baseada na figura em [PK79].

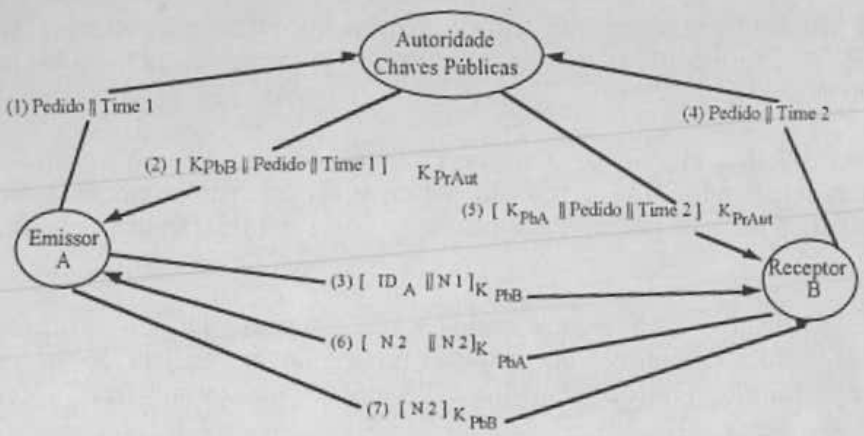

Figura 3.1 - Distribuição de chaves públicas por autoridades.

Para *A* iniciar uma comunicação autenticada com *B*, os seguintes passos são realizados:

1. O usuário *A* envia uma mensagem para a autoridade. A mensagem deve possuir um *timestamp* e um pedido para se comunicar com *B*.

2. A autoridade envia para *A* a chave pública de *B*, uma cópia do pedido original, e o *timestamp*. Todos esses elementos são criptografados com a chave privada da autoridade, e assim é assegurada a identidade do emissor. O *timestamp* garante que esta mensagem não é uma mensagem antiga da autoridade contendo uma outra chave pública de *B* que não seja a chave corrente. A cópia do pedido original permite que *A* verifique que o seu pedido não foi alterado durante a transmissão.

3. *A* envia uma mensagem criptografada para *B*. Esta mensagem deve conter um identificador de *A* e um desafio. Este desafio é um identificador qualquer para garantir que a mensagem não seja retransmitida.

4. *B* executa o passo (1) com a autoridade para obter a chave pública de *A*.

5. A autoridade executa o mesmo passo (2) com *B*.

6. *B* gera um novo desafio e envia para *A*, juntamente com o desafio que ele recebeu de *A*. Ambos são criptografados com a chave pública de *A*. *A* decripta esta mensagem e tem certeza de estar se comunicando com *B*.

7. *A* deve enviar para *B* o desafio gerado por *B*, criptografado com a chave pública de *B*. *B* decripta esta mensagem e tem a certeza de estar se comunicando com *A*.

Este protocolo apresenta algumas desvantagens. O protocolo contém sete mensagens, quatro das quais são utilizadas para a recuperação das chaves públicas no diretório através da autoridade. Este grande número de mensagens trocadas faz da autoridade um gargalo para o sistema. As mensagens 3, 6 e 7 têm por finalidade realizar a autenticação entre os usuários.

O usuário *A*, de posse da chave pública de *B*, pode se comunicar com *B* mais de uma vez sem a interferência da autoridade central, diminuindo o número de mensagens transmitidas na rede. Mas o processo de autenticação entre *A* e *B* ainda se faz necessário, para garantir que não se trata de um *replay* de uma mensagem antiga feita por um intruso.

Uma alternativa para a distribuição de chaves públicas sem a introdução de um gargalo no sistema é a utilização de uma autoridade de confiança junto com a técnica de certificados, o que faz surgir o emprego de uma *autoridade de certificação* (*certification authority* - CA), isto é, uma autoridade emissora de certificados de chaves públicas dos usuários. Neste esquema, cada chave pública gerada por um usuário deve possuir um certificado emitido pela CA.

O certificado tipicamente contém a chave pública do usuário, a identificação do usuário a quem o certificado diz respeito, a assinatura digital da CA realizada sobre as informações do certificado, a identificação da CA que gerou e assinou o certificado, além de outras informações, não relevantes para esta discussão. A identificação do usuário e da CA deve ser correspondente ao seu *distinguished name* (DN).

Uma autoridade de certificação é definida como uma autoridade a quem os usuários confiam a criação e assinatura de certificados [Ken93]. A expressão "distribuição de chaves públicas" é então substituída por "distribuição de certificados de chaves públicas".

O usuário deve registrar a sua chave pública, através de um procedimento de autenticação seguro. A partir deste registro, a autoridade gera um certificado. Um certificado pode ser considerado como uma mensagem gerada e assinada pela CA. Os certificados são usados para prover a componente pública autenticada dos usuários.

O uso de esquemas baseados em certificados garante que qualquer usuário possa ler um certificado para determinar a identificação e a chave pública de outro, e que qualquer usuário possa verificar que o certificado foi originado por uma CA e que não foi alterado, que somente a CA possa criar ou atualizar o certificado [PK79, Sta95].

Dados uma autoridade de certificação e um usuário, temos a seguinte notação:
- *C*: autoridade de certificação.
- *A*: usuário.
- $<<A>>$: certificado de *A*.
- $C<<A>>$: certificado de *A* gerado pela autoridade de certificação *C*.

A Figura 3.2 ilustra o novo esquema de distribuição de chaves públicas, onde é adotada a técnica de certificados, que são emitidos por uma CA de confiança dos usuários.

Neste esquema, cada usuário, após gerar o seu par de chaves, irá registrar a sua chave pública perante uma CA. Este processo é identificado pelas mensagens trocadas entre os usuários e a CA.

Quando *A* deseja se comunicar com *B*, basta *A* enviar o seu certificado para *B* e vice-versa. São os passos (1) e (2). Assim, quando *B* desejar autenticar o certificado de *A*, ele utiliza as informações contidas no certificado de *A*, para poder saber quem é a autoridade de *A*, e *B*, de posse da chave pública da CA, é capaz de verificar a autenticidade do certificado.

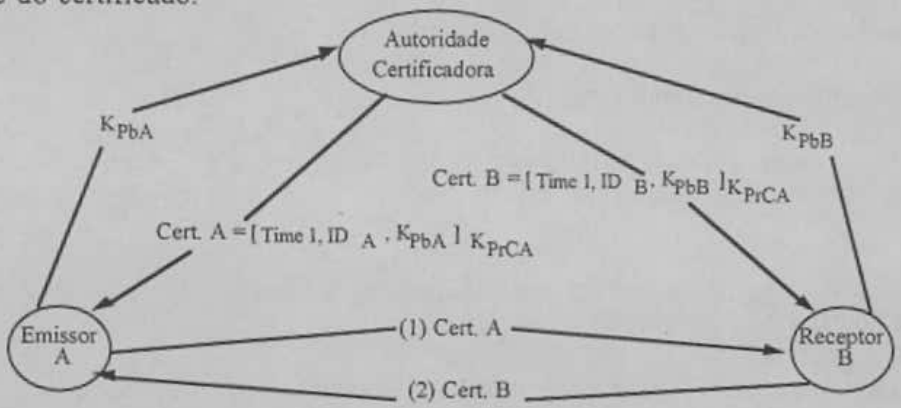

Figura 3.2 - Distribuição de certificados de chaves públicas.

Apesar deste esquema apresentar uma proposta de distribuição de chaves públicas mais segura que os anteriores, não inviabiliza ataques de intrusos.

Por exemplo, suponha que um intruso *C*, de posse da chave privada da autoridade, tenha gerado um certificado forjando ser *B*. Este certificado possui a chave pública de *C*, e é assinado com a chave privada da autoridade conhecida por *A* e *B*. O usuário *A*, utilizando o certificado falso, enviará mensagens para *B* que podem ser interceptadas e lidas por *C*. Estas mensagens, no entanto, não poderão ser decriptadas por *B*, pois a chave privada de *B* não corresponde à chave pública utilizada para criptografar a mensagem. Logo, se *C* não puder evitar que *B* receba a mensagem gerada por *A*, o seu comportamento será evidenciado.

Caso *C* possa interceptar a mensagem, ele pode decriptá-la, criptografá-la usando a chave pública de *B*, e enviar a nova versão para *B*. Quando *B* responde a *A*, o processo deve ser repetido. Isto envolve uma derivação clandestina da linha de teleprocessamento e remoções de mensagens, o que tipicamente acaba por evidenciar o ataque.

O número de mensagens trocadas entre os usuários e a autoridade neste esquema é menor, fazendo com que a CA não represente um gargalo para o sistema. O uso de certificados permite um retorno ao esquema de

difusão ou diretório, agora com distribuição de certificados e não de chaves. Uma vez que cada usuário conhece a chave pública da CA, ele pode verificar se qualquer certificado recebido é autêntico, sem a necessidade de comunicação com a autoridade.

Não ocorre a necessidade de autenticação mútua entre os usuários *A* e *B*, porque os usuários verificam a veracidade da chave pública através da assinatura contida no certificado.

4. Uso de autoridades de certificação em diretórios distribuídos

O uso de certificados de chave pública resolve o problema de falsificação de chaves, através da autenticação por assinatura digital, e permite que as chaves públicas sejam transmitidas através de canais inseguros. Surge então, o problema de disponibilizar os certificados de modo a torná-los de fácil acesso para os usuários do sistema, principalmente em ambientes distribuídos.

A utilização em conjunto da técnica de certificado, autoridade de certificação e diretórios distribuídos poderia representar uma solução a ser adotada para o problema. O diretório seria utilizado como repositório dos certificados de chaves públicas [Sta95, X592b].
Os serviços de diretório existentes são baseados na série de recomendações do X.500 [X.592a], da ITU-T, que define serviços de diretório que apresentam parte das funcionalidades de um banco de dados distribuído. Em particular, a recomendação X.509 [X.592b] define como diversas operações podem ser realizadas com segurança usando um canal de comunicação inseguro (como a Internet).

O diretório é um servidor ou um conjunto de servidores distribuídos que mantém as informações sobre usuários em um banco de dados. Assim, um diretório X.500 pode ser utilizado para armazenar, entre outras informações, certificados dos usuários.

A recomendação X.509, que é o padrão de autenticação, define o formato para a distribuição de chaves públicas e os protocolos de autenticação a serem utilizados. O X.509 é baseado no uso de criptografia assimétrica.

Dentro de uma comunidade como a Internet, não é prático que todos os usuários se submetam a apenas uma CA. Assim, pode ser mais prático ter mais de uma CA. O X.509 sugere que as autoridades de certificação sejam organizadas de uma maneira hierárquica, onde cada nível da hierarquia corresponde às autoridades de certificação pelo nível superior. Isto permite desenvolver uma espécie de navegação pela hierarquia, como será explicado mais adiante. A Figura 4.1 é um exemplo da hierarquia proposta.

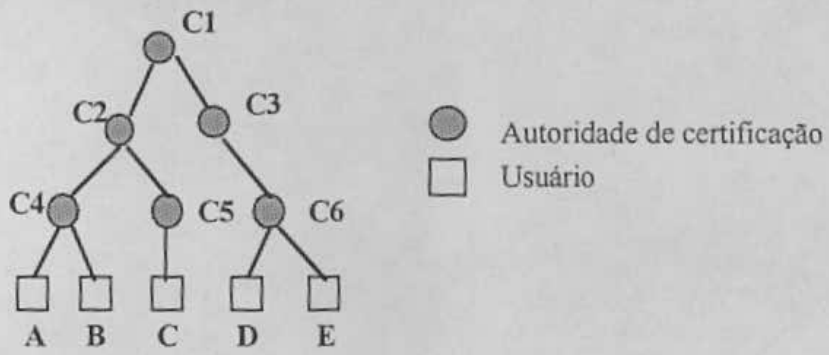

Figura 4.1- Hierarquia de certificação - X.509.

Agora suponha que um usuário *A* tenha obtido o seu certificado da CA *X1* e um usuário *B* tenha obtido o seu certificado da CA *X2*. Se *A* não possui a chave pública de *X2*, então o certificado de *B* gerado por *X2* não é de utilidade para *A*. *A* pode ler o certificado de *B*, mas não pode verificar a assinatura. Porém, se as duas autoridades de certificação trocarem suas chaves públicas de forma segura, o procedimento abaixo habilitará *A* a obter a chave pública de *B*.

1. *A* obtém do diretório o certificado de *X2* assinado por *X1*. A partir deste certificado, *A* pode obter a chave pública de *X2* através do certificado emitido por *X1*. Como *A* conhece a chave pública de *X1*, a autenticidade da chave de *X2* pode ser garantida.

2. *A* retorna ao diretório, e obtém o certificado de *B* assinado por *X2*, pois *A* agora conhece a chave pública de *X2*. A chave pública de *X2* é confiável, pois o seu certificado possui a assinatura de *X1*.

A usa um *caminho de certificação* (caminho de certificados) para obter a chave pública de *B*. De acordo com a notação X.509 temos:
X1 << X2 >> X2 << B >>.

Do mesmo modo, *B* pode obter a chave pública de *A* com o caminho inverso:

X2 << X1 >> X1 << A >>.

Pode-se estabelecer um caminho de certificação arbitrariamente longo de certificados das autoridades certificadoras, da forma:

X1 << X2 >> X2 << X3 >> X3 ... << Xn >> Xn << B >>.

Neste caso, cada par de autoridades de certificação do caminho de certificação (*Xi*, *Xi+1*) deve possuir certificados uma da outra. A Figura 4.2 ilustra o exemplo. Os círculos conectados indicam o relacionamento entre as autoridades; os círculos mais acima representam pontos mais altos na hierarquia. As notações ao lado dos círculos indicam os certificados mantidos no diretório por cada CA. Cada CA deve possuir dois tipos de certificados no diretório: o seu certificado e o certificado *reverso*.

O certificado reverso é o certificado emitido por uma CA para a CA imediatamente superior a ela. A exigência de que a cada CA corresponda um certificado e um certificado reverso permite o estabelecimento de caminhos de certificação entre quaisquer duas autoridades em uma hierarquia de certificação. Voltando ao exemplo da Figura 4.2, o certificado reverso é o que permite a validação da chave pública da CA *W* pela CA *X*.

Neste exemplo, o usuário *A* pode adquirir os certificados necessários, através do diretório, para estabelecer um caminho de certificação até *B*:

X << W >> W << V >> V << Y >> Y << Z >> Z << B >>.

A medida que *A* toma posse de cada certificado, ele vai descobrindo o caminho de certificação, onde o último certificado a ser inserido no caminho deve ser o da CA de *B*, que permite a verificação da autenticidade da chave pública de *B*. A chave pública de *B* é então verificada de modo confiável. Usando esta chave pública, *A* pode enviar mensagens criptografadas para *B* de modo seguro. Se *A* deseja receber mensagens criptografadas por *B*, ou assinar mensagens para *B*, então *B* requer a chave pública de *A*, que pode ser verificada pelo caminho de certificação:

Z << Y >> Y << V >> V << W >> W << X >> X << A >>.

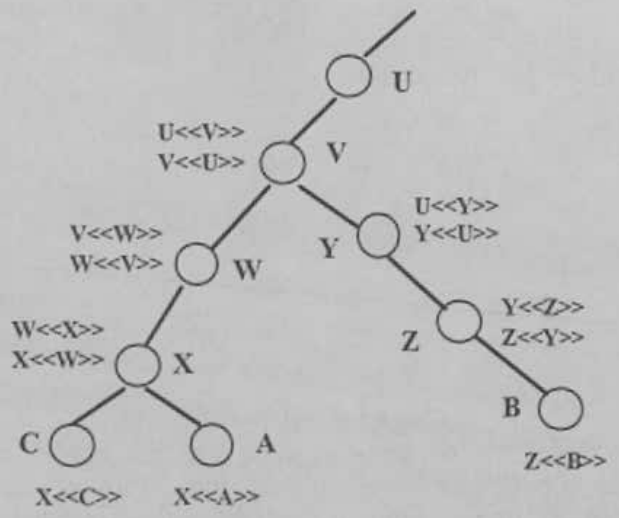

Figura 4.2 - Hierarquia de certificação - X.509.

Dentro deste panorama, cada CA fica responsável por uma fração do diretório, onde estarão armazenados os certificados emitidos por ela. Um usuário, ao pesquisar no diretório um determinado certificado de outro usuário, terá necessidade de estender a pesquisa para uma pesquisa de uma seqüência de certificados. Os certificados desta seqüência estão associados hierarquicamente de modo a formar um caminho de certificação.

5. Caminho de certificação

Nas seções 3 e 4 foram introduzidos os conceitos de certificado de chave pública, hierarquia de certificação e serviço de diretórios. O serviço de diretórios apresentou uma nova alternativa para o problema de distribuição de chaves públicas, pois ele define um serviço genérico implementando partes da funcionalidades de um banco de dados distribuído. Esta alternativa apresentou um novo problema a ser solucionado: encontar um caminho de certificação entre duas autoridades de certificação.

Esta seção descreve um algoritmo (BJB) e propõe dois novos algoritmos (HI e HH) para implementar uma solução para encontrar o caminho de certificação, analisando cada algoritmo em relação ao número de acessos ao serviço de diretório necessários. Também se descreve uma ferramenta que utiliza um destes novos algoritmos para a construção de caminhos de certificação.

Antes de iniciar a discussão sobre os algoritmos e do protótipo da ferramenta, faz-se necessário esclarecer conceitos já apresentados e apresentar novos conteitos para melhor entendimento dos algoritmos.

Uma hierarquia de certificação é composta por várias entidades, que podem ser usuários finais e autoridades de certificação. Para efeito de se encontrar o caminho de certificação, a hierarquia de certificação é modelada através de um grafo. Os usuários e as autoridades certificadoras são os nós deste grafo, e os certificados são os arcos. O certificado reverso é representado no grafo através de um arco em sentido contrário do certificado. O grafo contém um elo de um nó *N* para outro nó *M*, se e somente se, existe um certificado da CA representada por *N* emitido pela CA representada por *M*. Por exemplo, o grafo da Figura 4.1, mostra a existência de um certificado de *C2* assinado por *C1*, e de *C6* por *C3*.

A razão para a hierarquia ser modelada por um grafo e não por uma árvore, é que tanto as autoridades de certificação como usuários podem possuir certificados emitidos por mais de uma CA. Uma das razões para isto é a emissão de certificados entre autoridades de hierarquias de certificação originalmente independentes. Estes certificados são chamados de *certificados cruzados*. Um exemplo é ilustrado na Figura 5.1.

Os algoritmos utilizam os conceitos de *hierarquia de certificação* e *hierarquia de certificação completa*. A hierarquia de certificação de uma CA é composta por uma seqüência de autoridades de certificação, que vai desde a CA especificada até a CA "raiz" desta, isto é, uma autoridade de certificação que não é certificada por nenhuma outra. Uma hierarquia de certificação é dita completa, se ela contém uma hierarquia de certificação para cada uma das autoridades presentes nela.

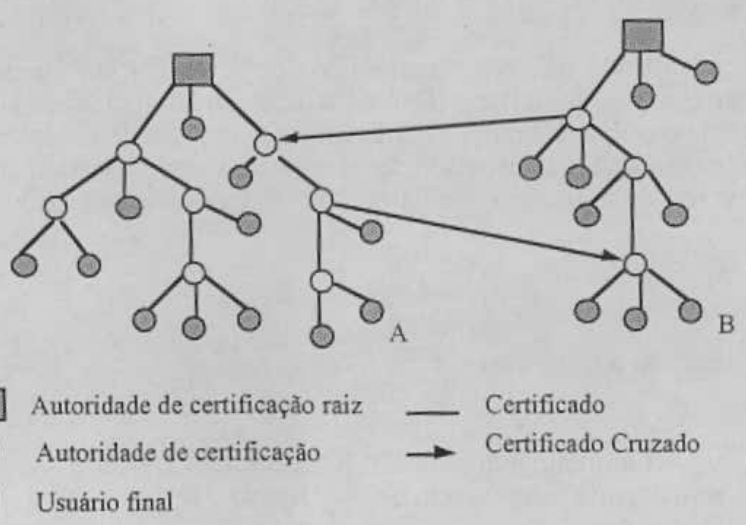

Figura 5.1 - Hierarquias de certificação ligadas por certificados cruzados.

Existem três possiblidades para se encontrar um caminho de certificação entre duas autoridades de certificação.

1. Construir uma hierarquia de certificação completa a partir da hierarquia do diretório. A partir desta hierarquia de certificação é possível extrair caminhos de certificação entre duas autoridades de certificação.

2. Construir duas hierarquias de certificação. As hierarquias seriam correspondentes às CA de origem e da CA de destino. Após a construção das duas hierarquias, pode ser encontrado um caminho de certificação entre duas autoridades de certificação, de origem e de destino. Neste caso, não é necessário construir a hierarquia de certificação completa.

3. Construir a hierarquia de certificação da CA de origem e dar um tratamento especial à construção da hierarquia de certificação da CA de destino. A construção da hierarquia de certificação da CA de destino é realizada até encontrar uma CA que pertença à hierarquia de certificação da CA de origem. Caso não exista uma CA comum às hierarquias, a construção da hierarquia de certificação para a CA de destino prossegue até chegar a uma autoridade certificadora raiz.

Estas possibilidades são discutidas a seguir com o auxílio dos algoritmos. Estes algoritmos são analisados de acordo com as requisições de construção de caminhos de certificação, onde a principal característica analisada é o número de acessos ao diretório.

5.1. Algoritmo BJB

Este algoritmo foi baseado em [BTK96]. O algoritmo é dividido em duas etapas. Na primeira etapa, uma base de dados é construída. Na segunda etapa, os K menores caminhos de certificação entre duas autoridades de certificação são encontrados.

Primeira etapa: Construir uma base de dados.
1. Criar um banco de dados local.
2. Pesquisar todas as autoridades de certificação existentes no diretório.
3. Armazenar no banco de dados local as informações obtidas da pesquisa acima, em especial os certificados.
4. Construir um grafo a partir das informações acima. Os nós do grafo são as autoridades de certificação e os arcos são os relacionamentos existentes entre elas, como descrito anteriormente.

Segunda etapa: Encontrar K menores caminhos entre duas autoridades de certificação.
1. Encontrar os K menores caminhos existentes entre duas autoridades de certificação. A seqüência de certificados que fazem parte dos caminhos são validados.

Este algoritmo gera um grafo que representa a hierarquia de certificação completa, contendo todas as autoridades de certificação existentes no diretório. A CA de origem, a CA de destino e o parâmetro K são os argumentos do algoritmo. O algoritmo utilizado para encontrar os K menores caminhos entre dois nós do grafo foi o algoritmo *Yen-Lawler* [Law76, Yen71].

5.2. Algoritmo HH

O algoritmo HH propõe a construção de duas hierarquias de certificação. Estas hierarquias são correspondentes à CA de origem e a CA de destino. Devido a esta característica, o algoritmo recebeu a denominação HH. Após a construção das duas hierarquias, é encontrado um caminho de certificação entre as duas autoridades, de origem e de destino. As raízes das hierarquias não são necessariamente a mesma. Este algoritmo é detalhado abaixo e faz uso do algoritmo H. O algoritmo HH é ilustrado na Figura 5.4.1.

```
Algoritmo HH ( CA_Origem, CA_Destino)
Inicio
  Criar GRAFO
        // grafo representado por listas de adjacências
  H( GRAFO, CA_Origem)
  H( GRAFO, CA_Destino)
  CAMINHO DE CERTIFICAÇÃO = Caminho_mínimo ( CA_Origem, CA_Destino)
        // se não existir uma lista de nós entre CA_Origem e CA_Destino
        // então não há caminho entre as duas CAs
Fim.
```

O algoritmo H faz uso de uma fila para auxiliar a construção do grafo. A fila tem o objetivo de armazenar as autoridades de certificação que farão parte do grafo. Isto é, quando não houver mais nenhuma CA na fila, o processo de construção do grafo termina.

```
Algoritmo H( GRAFO, CA)
Inicio
  Criar FILA
  Criar NÓ com CA
        // NÓ armazena dados da CA
  Inserir NÓ em GRAFO
  Inserir NÓ em FILA
  Enquanto FILA diferente de vazio Faça
```

```
    Remover elemento corrente de FILA
    Pesquisar CA correspondente ao elemento corrente
   Recuperar lista de certificados da CA pesquisada
    Para cada certificado pertencente a lista de certificados recuperada Faça
      Pesquisar a CA emissora do certificado em GRAFO
       // CAs emissoras dos certificados são os nós adjacentes a CA corrente
      Se não achou
        Criar novo NÓ com CA emissora
       // armazenar dados da CA emissora do certificado
        Inserir NÓ em FILA
        Inserir NÓ em GRAFO
      Fim-se
      Criar ARCO com CA emissora e CA
      Inserir ARCO em GRAFO
   Fim-para
  Fim-enquanto
Fim.
```

5.3. Algoritmo HI

Este algoritmo apresenta três pontos básicos de diferença em relação ao algoritmo HH:
1. Propõe a construção de apenas uma hierarquia de certificação (algoritmo H), que corresponde a uma única CA (de origem).

2. Um dos objetivos do algoritmo é encontrar uma CA que pertença às duas hierarquias de certificação. Assim, a construção da hierarquia de certificação da CA de destino recebe tratamento diferenciado (algoritmo I). Cada CA pertencente a esta hierarquia representa um novo nó a ser inserido no grafo. Mas, a cada inserção de um novo nó no grafo, é verificado se a CA correspondente já não pertence ao grafo. Em caso afirmativo, a construção da hierarquia de certificação da CA de destino termina, porque foi encontrada uma CA comum às duas hierarquias. Isto é, foi encontrado um ponto de interseção entre as duas hierarquias. Devido a esta característica, o algoritmo recebeu a denominação HI.

3. O caminho de certificação é composto por dois caminhos. Ele é composto pela união da seqüência de autoridades de certificação que formam um caminho da CA de origem até a CA comum, mais a seqüência de autoridades de certificação que formam um caminho da CA de destino até a CA comum.

```
Algoritmo HI ( CA_Origem, CA_Destino)
Inicio
  Criar GRAFO
        // grafo representado por listas de adjacências
  H( GRAFO, CA_Origem)
  I( GRAFO, CA_Destino, INTERSEÇÃO, CA_Comum)
  Se INTERSEÇÃO = Verdadeiro
     Caminho1 = Caminho_Mínimo( CA_Origem, CA_Comum)
        // retorna uma lista de nós
     Caminho2 = Caminho_Mínimo( CA_Destino, CA_Comum)
     CAMINHO DE CERTIFICAÇÃO = Caminho1 + Caminho2
  Senão
     "Não existe caminho de certificação entre CA_Origem e CA_Destino"
  Fim-se
Fim.

Algoritmo I( GRAFO, CA, INTERSEÇÃO, CA_Comum)
Inicio
  INTERSEÇÃO = Falso
  Criar FILA
  Pesquisar CA em GRAFO
  Se achou
    INTERSEÇÃO = Verdadeiro
  Senão
    Criar NÓ com CA
    Inserir NÓ em GRAFO
    Inserir NÓ em FILA
```

```
  Fim-se
  Enquanto FILA  diferente de vazio e INTERSEÇÃO = Falso
     Remover elemento corrente de FILA
        // corrente = NÓ
     Pesquisar CA correspondente ao elemento corrente
     Recuperar lista de certificados da CA pesquisada
     Para cada certificado pertencente a lista de certificados recuperada
                                        e INTERSEÇÃO = Falso faça
       Pesquisar a CA emissora do certificado em GRAFO
       Se achou
          INTERSEÇÃO = Verdadeiro
          CA_Comum = CA
       Senão
            Criar novo NÓ com CA emissora
                  // armazena os dados da CA emissora do certificado
            Inserir NÓ em GRAFO
            Inserir NÓ em FILA
       Fim-se
       Criar ARCO com CA emissora e CA
       Inserir ARCO em GRAFO
     Fim-para
  Fim-enquanto
Fim.
```

Caso não exista uma CA comum às duas hierarquias, não existe um caminho de certificação entre as duas autoridades, de origem e de destino, e a hierarquia de certificação da CA de destino possuíra uma CA raiz. O algoritmo HI, ilustrado na Figura 5.4.2, faz uso do algoritmo H descrito anteriormente, e do algoritmo I. O algoritmo I faz uso de uma fila com o mesmo propósito descrito no algoritmo H.

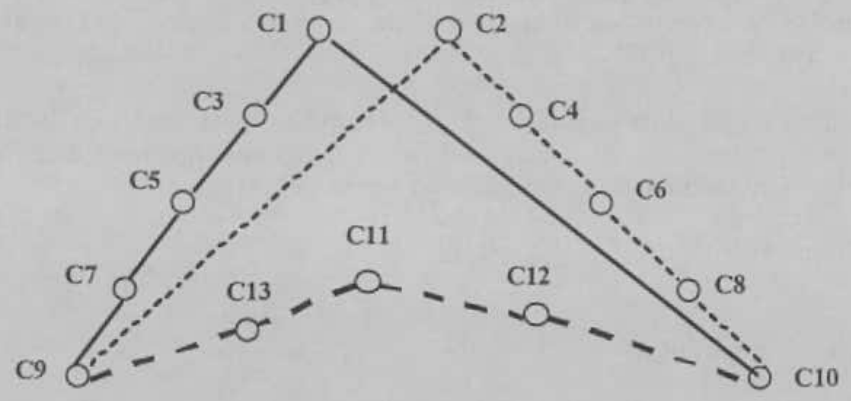

Figura 5.3.1 - Menor caminho de certificação.

O caminho de certificação encontrado não necessariamente é o menor caminho existente entre duas autoridades de certificação. Há uma possibilidade de que o grafo representando a hierarquia de certificação construída, não represente a hierarquia de certificação completa. Um exemplo é ilustrado na Figura 5.3.1. Neste exemplo, o menor caminho entre *C9* (origem) e *C10* (destino) é composto por *C13*, *C11* e *C12*. Mas o caminho de certificação encontrado pelo algoritmo é composto por *C7*, *C5*, *C3* e *C1*. Voltaremos a este ponto na próxima seção.

5.4. Algoritmos BJB, HH e HI

Nos algoritmos descritos nas seções 5.2 e 5.3, o algoritmo usado para encontrar o menor caminho entre dois nós do grafo foi o algoritmo de *Dijkstra* [Law76]. A seqüência de certificados que fazem parte do caminho de certificação é validada a medida que cada certificado é representado no grafo, ou seja, a cada inclusão de um arco. Caso um certificado seja invalidado, o arco correspondente não é inserido no grafo.

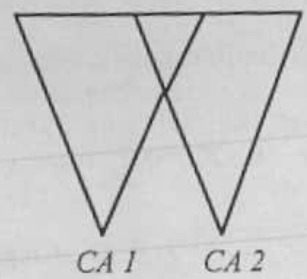

Figura 5.4.1 - Algoritmo HH.

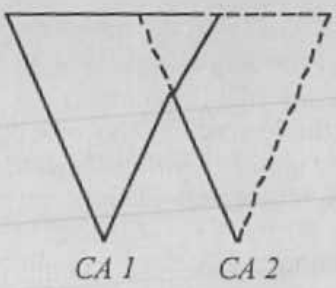

Figura 5.4.2 - Algoritmo HH e HI.

Comparando-se os algoritmos, observa-se que o HI realiza menos acessos ao diretório, podendo ser o mais eficiente. Isto porque espera-se que o algoritmo recupere apenas parte da hierarquia de certificação da CA de destino. O preço desta provável diminuição do número de acessos ao diretório é o fato de não se encontrar necessariamente o menor caminho, como discutido na seção anterior.

O grafo com a hierarquia de certificação completa construído pelo algoritmo BJB pode ser aproveitado em buscas subsequentes de caminhos de certificação. O emprego do algoritmo HI é interessante em um padrão de uso onde as requisições de construção de caminhos não tenham uma frequência que justifique a manutenção local de um espelho da hierarquia de certificação completa.

5.5. A ferramenta PathFinder

O protótipo de uma ferramenta foi construído com o objetivo de realizar a pesquisa automática de um caminho de certificação dentro de uma hierarquia de autoridades de certificação, tendo-se apenas o conhecimento da CA de origem e da CA de destino, e utilizando-se um serviço de diretório como repositório de certificados de chaves públicas.

A implementação do PathFinder utiliza o algoritmo HI para encontrar o caminho de certificação. A implementação foi feita no ambiente UNIX. Foram utilizadas funções de criptografia que implementam o algoritmo RSA [Sch94]. O sistema também apresenta um esquema de armazenamento local, a fim de que possam ser armazenadas informações das autoridades de certificação pesquisadas no diretório. Isto é feito numa tentativa de se fazer menos acessos ao diretório, aumentando a eficiência do sistema. De tempos em tempos este *cache* do diretório deverá ser atualizado.

O serviço de diretório no sistema é representado pelo SLAPD [Uni96], que é um servidor local. Esta configuração é ilustrada na Figura 5.5.1. O SLAPD é um servidor que funciona *stand alone* oferecendo os serviços de diretório independente de qualquer servidor de diretório existente. Este servidor é uma implementação do protocolo LDAP (*lightweight directory access protocol*) [LDA]. O LDAP foi proposto para oferecer um subconjunto das funcionalidades do X.500, sem a necessidade do uso da pilha de protocolos completa que este supõe.

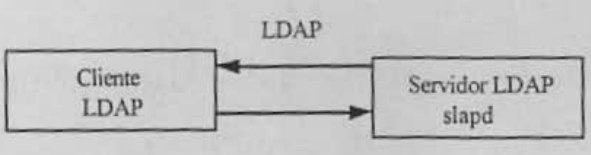

Figura 5.5.1 - Servidor local slapd.

O LDAP e o servidor SLAPD foram escolhidos para a implementação desta ferramenta por sua facilidade de uso e instalação em comparação ao QUIPU [Qui] (implementação mais conhecida do serviços de diretório definidos pelo X.500). O conjunto de operações de acesso ao diretório oferecido pelo sistema era suficiente para as nossas necessidades de desenvolvimento. Além disto, estas operações são oferecidas na forma de uma biblioteca de rotinas, o que facilita em muito a construção de clientes do diretório [HKW95, How95, HS95].

6. Considerações finais

Esse trabalho teve início com um estudo sobre sistemas de correio eletrônico com serviços de segurança disponíveis na Internet. Várias propostas e implementações desse tipo de sistema tem surgido nos últimos anos [Zim93], [Rio93], [Sch95], [Bal93], [Ken93], [Kal93] e [Lin93]. Com pequenas variações, esses sistemas oferecem serviços de confidencialidade e assinatura digital, baseando-se no esquema híbrido de criptografia simétrica e assimétrica apresentado na Figura 2.1, que vem se mostrando bastante satisfatório.

O problema que ainda não está bem resolvido é a questão de distribuição confiável de chaves públicas. O sistema em maior uso na Internet atualmente, o PGP, basicamente deixa a definição do esquema de distribuição a cargo do usuário do sistema. Nas propostas de padrões Internet [Ken93], supõe-se a existência de uma hierarquia de autoridades de certificação com uso de serviços de diretório X.500, mas até o momento não foi definido um tal conjunto de autoridades.

Procurou-se com este trabalho melhorar a compreensão do problema de distribuição de chaves públicas, através da discussão de alternativas e problemas associados. O desenvolvimento do protótipo contribuiu para o melhor entendimento das dificuldades associadas a construção de um esquema de distribuição de chaves, baseado em autoridades de certificação e serviço de diretórios, adequado para redes geograficamente distribuídas. Em particular, a discussão de possíveis soluções para a construção de caminhos de certificação nos parece importante, principalmente pela ausência de trabalhos sobre o assunto (uma exceção é [BTK96]) e pela necessidade da existência de um esquema que permita troca de chaves públicas entre entidades de modo confiável.

Para permitir a continuidade do trabalho descrito aqui, desenvolveu-se [CI96] uma implementação do PEM que foi integrada ao agente de usuário MH [Pee95] para ambiente UNIX. O sistema usa, de forma transparente para o usuário, o serviço de diretórios LDAP para procurar a chave pública do destinatário de uma mensagem. Um próximo passo será a integração da ferramenta descrita neste trabalho a este sistema. Pretende-se experimentar a utilização do sistema resultante no ambiente composto pelos diversos laboratórios associados ao DI da PUC-Rio. Isto implicará também na instalação efetiva de um sistema de diretórios distribuído, não realizado para a construção da primeira versão descrita aqui.

O protótipo desenvolvido permite acessos ao diretório distribuído, desde que esta configuração exista. O uso de um servidor LDAP como *front-end* para o X.500 seria uma solução que tornaria o acesso ao diretório distribuído transparente para a ferramenta proposta. Uma outra possibilidade é fazer que a ferramenta seja capaz de tratar respostas onde o servidor de diretório retorna uma referência para um novo servidor a ser consultado.

7. Agradecimentos

Este trabalho foi parcialmente apoiado pela Capes (bolsa de mestrado, primeiro autor) e pelo CNPq (bolsa de produtividade em pesquisa 300803/93-1, segundo autor).

8. Bibliografia

[Bal93] David Balenson. Privacy Enhancement for Internet Eletronic Mail: Part III: Algoritms, Models, and Identifiers. RFC 1423, Internet Engineering Task Force (IETF), February 1993.

[BTK96] B. Jerma-Blazic, D. Trcek, T. Klobuc, and F. Bracun. A tool for suport of key distribution and validity certificate check in global Directory service. Computer Networks and ISDN Systems, 28:709-717, January 1996.

[CI96] Simone Carrocino e Alexandre Ingles. Correio Eletrônico com Privacidade e Autenticação. Trabalho final de curso. DI/PUC-Rio, 1996.

[DH76] Whitfield Diffie and M. E. Hellman. New Directions in Cryptography. IEEE Transactions on Information Theory, 22(6):644-654, November 1976.

[Dif88] Whitfield Diffie. The First Ten Years of Public Key Cryptology. In Proceedings of the IEEE, pages 560-577, May 1988.

[HKW95] Tim Howes, S. Kille, and W. Yeong. Lightweight Directory Access Protocol. RFC 1777, Network Working Group, March 1995.

[How95] Tim Howes, The Lightweight Directory Access Protocol: X.500 Lite. Technical report, University of Michigan, July 1995.

[HS95] Tim Howes and Mark Smith. The LDAP Application Program Interfaces. RFC 1823, Network Working Group, August 1995.

[Kal93] B. Kaliski. Privacy Enhancement for Internet Eletronic Mail: Part IV: Key Certification and Related Services. RFC 1424, Internet Engineering Task Force (IETF), February 1993.

[Ken93] Stephen Kent. Privacy Enhancement for Internet Eletronic Mail: Part II: Certificate-Based Key Management. RFC 1422, Internet Engineering Task Force (IETF), February 1993.

[Lin93] John Linn. Privacy Enhancement for Internet Eletronic Mail: Part I: Message Encryption and Authentication Procedures. RFC 1421, Internet Engineering Task Force (IETF), February 1993.

[KPS95] C. Kaufman, R. Perlman, and M. Speciner. Network security: privace communications in a public world. Printice-Hall, New Jersey, 1995.

[Law76] E. L. Lawler. Combinatorial Optimization: Networks and Matroids. Hold, Rinehart and Winston, 1976.

[LDA] LDAP - Lightweight Directory Access Protocol. Http//www.umich.edu/~rsug/ldap.

[Luc85] Cláudio L. Luchhesi. Introdução à criptografia computacional. Editora da UNICAMP, Campinas, 1985.

[Nec92] James Nechvatal. Public Key Cryptography. Library os Congress Cataloging-in-Publication Data, pages 177-287, 1992.

[Pee95] Jerry Peek. MH & xmh: Email for Users & programmers. O'Reilly & Associates, 1995.

[PK79] G. J. Popek and C. S. Kline. Encryptation and Secure Computer Networks. Computers & Security, 5:243-250, 1986.

[Qui] Quipu. ftp//uk.isode.com.

[Rio95] Mark Riordan. RIPEM User Guide, March 1995.

[Sch94] B. Schneier. Applied cryptography: protocols, algorithms, and source code in C. John Willey & Sons, Inc. New York, 1994.

[Sch95] W. Schneider. SecuDE. Institut fur TeleKooperations Technik, March 1995. Vol. I and III.

[Sta95] William Stallings. Network and internetwork security: principles and pratice. Printice-Hall, New Jersey, 1995.

[Uni96] University of Michigan. The SLAPD and SLURPD Administrator's Guide, April 1996. Release 3.3.

[X592a] CCITT Recommendations X.500 - The Directory: Overview of Concepts, Models, and Services, April 1992, Geneva.

[X592b] CCITT Recommendations X.509 - The Directory: Authentication Framework, April 1992, Geneva.

[Yen71] J. V. Yen. Finding the K shortest loopless paths in a network. Management Science, 17:712-716, 1971.

[Zim93] Philip Zimmermann. PGP User's Guide, June 1993, Vol. I and II.