Teias de federações como extensões ao modelo de autenticação e autorização SDSI / SPKI

Altair Olivo Santin[1,3], Joni da Silva Fraga[1], Emerson R. de Mello[1], Frank Siqueira[2]

[1]Departamento de Automação e Sistemas e [2]Departamento de Informática e Estatística - Universidade Federal de Santa Catarina (UFSC)
CEP: 88040-900 – Florianópolis – SC – Brasil

[3]Programa de Pós-Graduação em Informática Aplicada - Pontifícia Universidade Católica do Paraná (PUCPR) – CEP: 80.215-901 – Curitiba - PR – Brasil

{santin, fraga, emerson}@das.ufsc.br, frank@inf.ufsc.br

Abstract. *Classic security systems use a trust model centered in the authentication procedure, which depends on a naming service. Even when using a Public Key Infrastructure as X.509, such systems are not easily scalable and can become single points of failure or performance bottlenecks. Newer systems, with trust paradigm focused on the client and based on authorization chains, as SDSI/SPKI, are much more scalable. However, they offer some difficulty on locating the chain linking the client to a server. This paper defines extensions to the SDSI/SPKI authorization and authentication model, which allow the client to build new chains in order to link it to a server when the corresponding path does not exist.*

Resumo. *Os sistemas clássicos com modelo de confiança centrado na autenticação, vinculada ao serviço de nomes, mesmo através do uso de "Public Key Infrastructure", como a X.509, podem torna-se o ponto único de falha, vulnerabilidades ou de comprometimento do desempenho do sistema. Sistemas mais recentes, com paradigma de confiança focado no cliente e baseado em cadeias de autorização, como o SDSI / SPKI, sofrem com a dificuldade de localização da cadeia que liga um cliente a um servidor. Com as extensões, a autorização e a autenticação SDSI / SPKI propostas neste trabalho, uma alternativa para a formação de novas cadeias é oferecida aos clientes para ligá-los a um servidor quando este caminho não existe.*

1. INTRODUÇÃO

Na visão clássica da autenticação e autorização em sistemas distribuídos, o serviço de nomes centraliza a autenticação restringindo a sua atuação ao domínio de nomes local. Por sua vez, a autorização geralmente apresenta seus controles implementados de forma distribuída. Este modelo, usado normalmente em redes corporativas, assume uma certa complexidade quando o ambiente é a rede mundial. Para vencer as limitações de escala, é necessário que sejam criadas relações de confiança interdomínios para cobrir um espaço de nomes global. Nestas condições, a gerência e a operacionalização destas relações, em geral, tornam-se tarefas árduas.

As PKI's (*Public Key Infrastructure*) tendem a oferecer facilidades de efetivação da autenticação num contexto global. A PKI X.509, por exemplo, adota um sistema de nomeação global (X.500), que baseado num modelo de confiança hierárquico formado por

CA's (*Certification Authorities*) permite a construção de cadeias (*chains*) de autenticação. Neste modelo, as cadeias de autenticação iniciam numa CA raiz (*root*) e terminam em um usuário (principal). Embora seja a PKI mais utilizada atualmente, o modelo global X.509 enfrenta dificuldades de adequação a legislação local e autônoma de cada país e de utilização devido à rigidez e complexidade de seu esquema.

Em outras palavras, o modelo de confiança (*trust model*) baseado em uma entidade centralizadora (serviço de nomes / autenticação), além de propiciar a criação de pontos críticos em relação à falhas e a vulnerabilidades, impõe restrições ao desempenho e a escalabilidade do sistema em ambientes de larga escala [Horst e Lischla, 2001].

Quando consideradas aplicações na Internet, a autenticação e a autorização devem evoluir tomando como base modelos onde às relações de confiança possam ser estabelecidas de maneira distribuída, escalável e flexível. O *Pretty Good Privacy* (PGP), utilizado na cifragem e autenticação de arquivos e correio eletrônico [Garfinkel, 1995], adota uma estrutura para gerenciamento e certificação de chaves que está baseada numa teia de confiança (*web of trust*). Se comparadas às hierarquias X.509, as teias de confiança PGP – construídas de maneira arbitrária – são bastante flexíveis e apresentam-se como mais adequadas às características da rede mundial. Porém, a adoção deste modelo baseado em ponderações, cria dificuldades na tomada de decisões de confiança por exigir múltiplas assinaturas num mesmo certificado para dar-lhe credibilidade.

Nesta linha de modelos de confiança igualitários que visam à adequação ao ambiente distribuído da rede mundial (a Internet), o conceito de gerência de confiança foi proposto como fundamento para paradigmas focados principalmente na autorização [Blaze et al., 1996]. A gerência de confiança unifica as noções de política de segurança, credenciais, controle de acesso e autorização.

Na literatura técnica científica são encontradas duas abordagens distintas que seguem este conceito. Na primeira, a gerência de confiança é efetivada usando uma linguagem na descrição da autorização e das credenciais, e um motor-lógico (*engine*) definindo o comportamento do módulo de checagem de conformidade (*compliance checker*); o *PolicyMaker* e o *KeyNote* [Blaze et al., 1999] são sistemas que adotam esta abordagem. O conceito de gerência de confiança pode também ser implantado com o uso de uma estrutura de dados padronizada permitindo a descrição tanto de credencias que definem a autorização como de políticas de segurança; o *Simple Distributed Security Infrastructure / Simple Public Key Infrastructure* (SDSI / SPKI) é um exemplo desta abordagem. Em ambos os modelos, checagens de conformidade indicam a adequação das credenciais apresentadas pelos principais às políticas de segurança locais – as quais são especificadas em guardiões (monitores de referência) distribuídos no sistema.

A infra-estrutura SDSI / SPKI foi motivada pela percepção da complexidade do esquema global de nomeação do X.509. O SDSI [Lampson e Rivest, 1996] é uma infra-estrutura de segurança que tem como objetivo principal facilitar a construção de sistemas distribuídos seguros e escaláveis. O SPKI [Ellison *et al.*, 1999] é o resultado dos esforços concentrados no projeto de um modelo de autorização simples, implementável e bem definido. Por apresentarem propósitos complementares, as propostas SPKI e SDSI são combinadas, formando uma base única para a autenticação e autorização em aplicações distribuídas.

A principal dificuldade do SDSI / SPKI é encontrar a cadeia de autorização comprovando que um principal (cliente) possui as permissões para o acesso desejado a um objeto / serviço no sistema distribuído. Várias experiências foram publicadas mostrando propostas de arquiteturas e algoritmos para auxiliarem nas buscas das cadeias de certificados. Porém, quando a busca resultava em insucesso, em nenhuma das propostas foram sugeridas alternativas e o processo de busca apenas se encerrava.

Neste trabalho é mostrado o uso de cadeias de confiança na efetivação do processo de autenticação e autorização em sistemas distribuídos de larga escala. O modelo de confiança do SDSI / SPKI é estendido através da adoção do conceito de federações no sentido de facilitar a gestão de certificados e também a construção de novas relações de confiança em sistemas de larga escala. Federações definem domínios de relações de confiança. Teias de federações estabelecem mecanismos que permitem aos principais compor relações de amplitude global. Assim, por exemplo, na inexistência da cadeia, os principais podem pesquisar certificados na teia e depois negociar a concessão dos privilégios demandados para formar uma nova cadeia de autorização.

O restante do texto está organizado da seguinte maneira: a seção 2 revisa brevemente SDSI / SPKI; a seção 3 introduz as extensões ao modelo de autenticação e autorização do SDSI / SPKI; a seção 4 aborda a formação de novas cadeias de autorização; a seção 5 expõe aspectos de implementação da arquitetura; a seção 6 apresenta os trabalhos relacionados; e por fim, a seção 7 relaciona as conclusões do trabalho.

2. SDSI /SPKI

A infra-estrutura SDSI / SPKI apresenta um modelo de confiança – baseado em cadeias de autorização – bastante simples e flexível. O sistema de nomeação herdado do SDSI induz uso de nomes locais para principais[1], mesmo no sentido global de um ambiente distribuído. Ou seja, no lugar de criar um espaço de nomes global único, os nomes SDSI / SPKI são sempre locais.

O SDSI / SPKI define um modelo de confiança igualitário: os principais são chaves públicas que podem assinar e emitir certificados da mesma forma que a CA no X.509. Na atual versão do SDSI / SPKI são definidos dois tipos distintos de certificados: um para nomes e outro para autorização.

- Certificado de nome

Os certificados de nome (Tabela 1-a) ligam nomes a chaves públicas ou, ainda, a outros nomes. Os nomes que integram estes certificados apresentam seus significados limitados ao espaço de nomes do emissor do certificado. A concatenação da chave pública do emissor do certificado com um nome local representa um identificador global único no *SDSI / SPKI*. O emissor de um certificado é sempre uma chave pública.

Assim, qualquer principal pode criar seu par de chaves (privada e pública) e então, associar um nome à chave pública do par – que é divulgado através de um certificado de nome. Um certificado de nome pode fazer referência a um outro nome (publicado num certificado de nome por outro principal) de modo a formar cadeias de nomes através do encadeamento de certificados (*linked names*).

[1] Entidades ativas que possuem um par de chaves (privada e pública) e são capazes de executar assinaturas digitais.

No SDSI / SPKI os nomes e as cadeias de nomes são usados apenas para facilitar a busca dos verdadeiros identificadores de principais: as chaves públicas. Quando nomes precisam ser "resolvidos", a cadeia de nomes ligados deve ser percorrida até chegar à chave pública correspondente. O processo de recuperação da cadeia de nomes para alcançar o certificado de nome original é chamado de "redução da cadeia de nomes".

Tabela 1 - Formato dos certificados de nome (a) e de autorização (b) SPKI

Campos	Descrição
Issuer	Chave pública que está definindo o "Name" em seu espaço de nomes local.
Name	Nome local que está sendo atribuído ao sujeito.
Subject	Chave pública ou nome definido em outro espaço de nomes e que será redefinido no espaço de nomes local ao emissor.
Validity dates	Especifica o período de validade – em formato 'data-hora'.

(a)

Campos	Descrição
Issuer	Chave pública do emissor do certificado.
Subject	Chave pública (ou *hash* dessa) ou nome identificando o principal que receberá a autorização.
Delegation	Valor lógico indicando se o sujeito pode (*True*) ou não (*False*) propagar a autorização que lhe foi delegada pelo emissor.
Authorization	Especifica as permissões concedidas pelo emissor.
Validity dates	Especifica o período de validade – em formato 'data-hora'.

(b)

- Certificado de autorização

Os certificados de autorização SDSI / SPKI (Tabela 1-b) ligam autorizações a um nome, a um "grupo especial" de principais (*threshold subjects*) ou a uma chave pública. Através destes certificados, o emissor pode delegar permissões de acesso a outros principais (outras chaves públicas) no sistema. Na arquitetura SDSI / SPKI um primeiro certificado de autorização é emitido a partir das ACL's do guardião, definindo algumas permissões de acesso ao recurso protegido. Delegações sucessivas de um mesmo conjunto de permissões formam uma cadeia de certificados de autorização que partem da chave pública responsável pelo guardião do recurso. Qualquer chave pública pode emitir um certificado de autorização, seja definindo um conjunto de permissões ou mesmo delegando permissões recebidas anteriormente.

Na verdade, no que se refere a certificados de autorização e ACL's, o SDSI / SPKI define um formato único de representação (Tabela 1-b), facilitando as atribuições, delegações e checagens de autorização. O conteúdo do certificado pode ser o mesmo da ACL, porém, ao certificado é acrescido o campo do emissor que vai assinando-o; a ACL não possui este campo porque é local ao guardião do serviço. Para efeito de verificação de autorização (redução da cadeia) "o campo de emissor na ACL" é preenchido com a palavra reservada '*SELF*' (Figura 1).

Os certificados de autorização SDSI / SPKI podem ser utilizados para dois propósitos distintos. Em um, quando o *bit* de delegação estiver desligado (delegação não permitida), os atributos de privilégio não podem ser repassados. Neste caso, o sujeito (principal) deve guardar o certificado de autorização como sendo do tipo "privado", do qual só o próprio sujeito pode fazer uso. Quando o *bit* de delegação estiver ligado (delegação permitida), o sujeito está de posse de um certificado de autorização do tipo "público", do qual o mesmo pode fazer o uso que bem entender. Ou seja, o principal pode guardá-lo para seu uso particular, repassá-lo a terceiros na íntegra ou num subconjunto – quando isto se fizer necessário, ou ambos [Gasser e McDermott, 1990].

É importante considerar que certificados de autorização públicos não transportam apenas atributos de privilégio, mas a confiança do emissor no sujeito – na administração dos atributos de privilégio concedidos. Uma vez concedida uma autorização (delegável ou não), esta é irrevogável e limitada apenas às restrições temporais

especificadas no campo de validade do certificado. Esta confiança também não é necessariamente irrestrita, podendo prever limitações ditadas por outras condições codificadas no certificado. Para efeito de controle de acesso os direitos concedidos através das delegações sucessivas (cadeias de autorização), devem ser "reduzidos / resumidos" em um certificado único contendo os direitos delegados até um sujeito, num processo denominado de "redução da cadeia de autorização".

A Figura 1 ilustra o fluxo de autorização baseado no modelo confiança SDSI / SPKI. Através da delegação de privilégios, a partir do servidor de aplicação, são criadas cadeias (caminhos) de autorização que formam a rede de confiança entre o servidor e o cliente. Na figura, os clientes *A* e *B*, após receberem os certificados indicados, terão suas cadeias de autorização necessárias para o acesso ao servidor. As cadeias de autorização são normalmente construídas de maneira arbitrária, cabendo ao possuidor dos privilégios guardar os certificados e apresentá-los ao servidor quando das requisições de acesso correspondentes. Baseado nisto, pode-se acrescentar que o modelo de confiança adotado no SDSI / SPKI é focado no cliente.

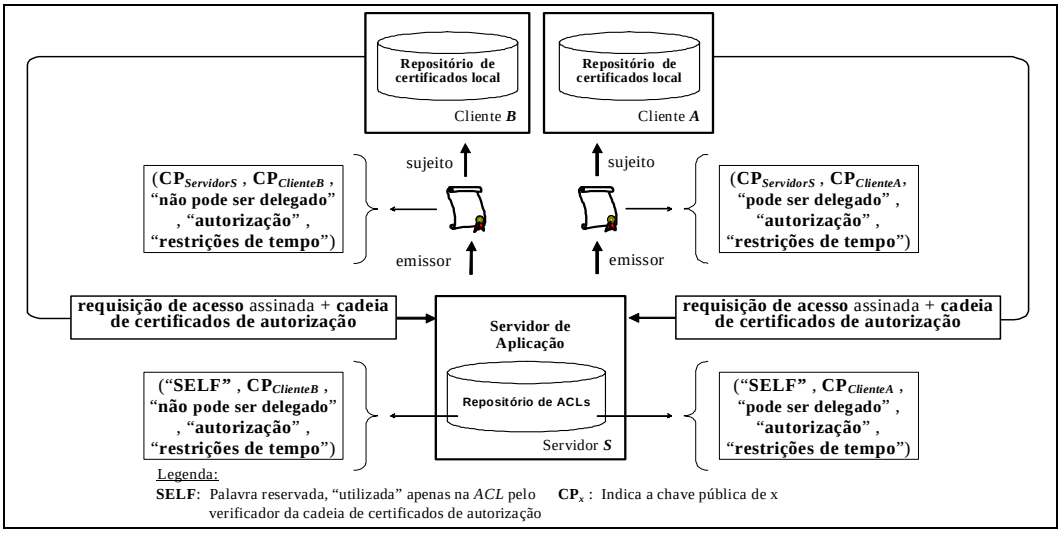

Figura 1 – Fluxo de autorização em SDSI / SPKI (modelo de confiança "focado no cliente")

O fluxo de autorização mostrado na Figura 1 permite a descentralização da administração na concessão de permissões de acesso – nos casos em que o servidor emite certificados permitindo a delegação de privilégios. Esta descentralização caracteriza uma flexibilidade maior do SDSI / SPKI sobre outras infra-estruturas de chaves públicas que seguem modelos de confiança convencionais.

3. PROPOSIÇÃO DE UM MODELO DE CONFIANÇA: EXTENSÕES AO SDSI / SPKI

Nesta seção são considerados aspectos referentes a uma proposta de modelo de confiança que permite definições de controles de autorização, autenticação e a formação de novas cadeias de autorização que na verdade é uma extensão do modelo de confiança do SDSI / SPKI.

O modelo de confiança proposto está fundamentado na noção de "federação", que enfatiza o agrupamento de principais com interesses afins. As federações têm o objetivo de

auxiliar os seus congregados na redução de nomes de principais e na construção de novas cadeias de autorização, através de seu gerente de certificados (GC).

Através do processo de filiação a federação, os principais passam a ter acesso aos serviços oferecidos pela mesma e novas relações de confiança entre estes principais podem ser estabelecidas. Neste sentido, o modelo de confiança SDSI / SPKI é estendido através da agregação do GC, oferecendo uma alternativa para a busca de certificados tanto para a resolução de nomes quanto para construção de novas cadeias de autorização.

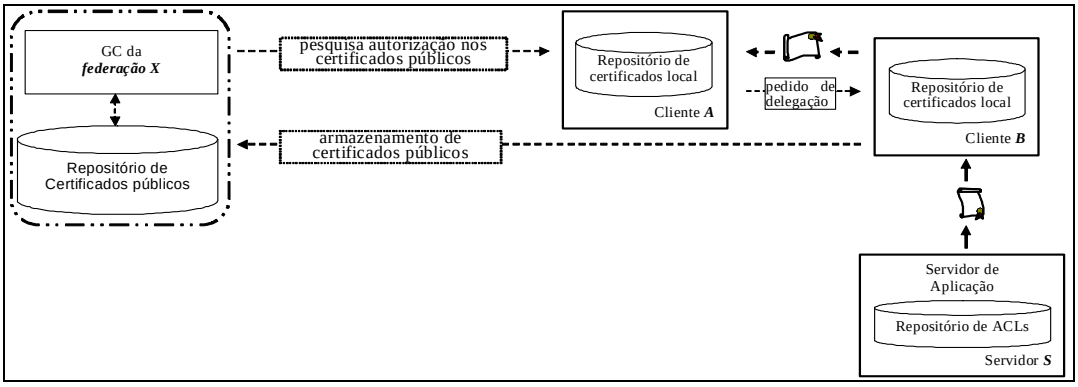

Figura 2 – Modelo de Confiança SDSI / SPKI Estendido

A Figura 2 mostra o GC sendo agregado ao modelo clássico do SDSI / SPKI e permitindo que o cliente *B* armazene os seus certificados públicos no repositório da federação. Através de uma pesquisa no repositório do GC, o cliente *A* – que não tem acesso ao servidor *S* – pode identificar na federação um principal (o cliente *B*) possuidor de tal privilégio e, então, negociar a concessão do privilégio com o mesmo.

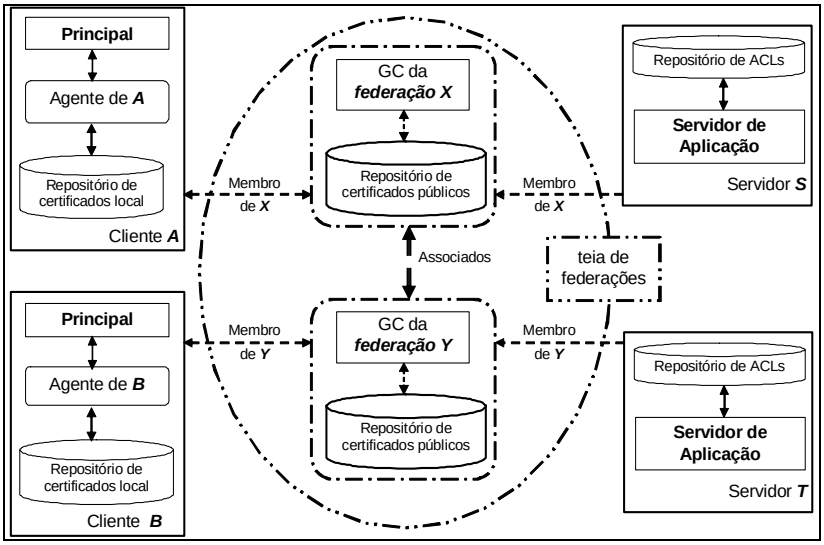

Figura 3 – Visão geral da teia de federações

As filiações de um cliente a várias federações permitem ao mesmo o acesso facilitado aos certificados de autorização públicos dos membros destas federações. Porém, o número de filiações necessárias para ter um determinado nível de presença na rede mundial pode caracterizar também um problema de escalabilidade. Os requisitos de escala são conseguidos no modelo proposto através de associações de federações. Os gerentes de certificados associam-se uns aos outros – àqueles que, por afinidade, melhor representem

as necessidades de seus membros – através de relações de confiança formando teias de federações (na Figura 3, o GC da federação *X* se associa ao GC da federação *Y*). Assim, os clientes e os servidores estariam eximidos da necessidade de filiar-se a um número considerável de federações para ter escopo global.

A Figura 3 ilustra ainda como estão organizadas as entidades que compõem a teia de federações. Os certificados de autorização (privados e públicos) de um cliente são mantidos em um repositório local sob a responsabilidade de um agente operando em nome deste principal em seu domínio local. Clientes fazem disponíveis certificados de nomes emitidos por seus principais e seus certificados de autorização públicos nos GCs das federações em que participam. Estes certificados disponíveis através dos GCs são usados na procura de potenciais emissores de permissões delegáveis.

Pode-se observar na proposta que não há nenhum tipo de centralização ou hierarquização. As teias de federações que são formadas arbitrariamente não desempenham nenhum papel ativo nas cadeias de autorização, ou seja, desempenham apenas funções de apoio (suporte) nos processos de autorização.

Uma federação, portanto, é composta basicamente de três entidades: clientes, servidores (de aplicação) e gerente de certificados, que serão detalhados nos tópicos a seguir.

3.1 CLIENTES E SERVIDORES DE APLICAÇÃO

O cliente representa o principal que cria certificados de nome, propaga os certificados de autorização por delegação, participa de *threshold certificates*, emite requisições de acesso e participa da formação de novas cadeias.

O armazenamento e a recuperação de certificados no espaço de nomes do cliente são concretizados a partir de seu agente (Figura 3). O agente corresponde a um software executando operações de gestão dos certificados disponíveis no repositório local, que compreendem a verificação e a efetivação de assinaturas, pesquisas de cadeias de certificados, negociações na delegação de permissões, emissão de novos certificados de autorização e manutenção da consistência dos nomes locais. O agente está sempre ativo no sistema e o cliente se comunica com o mesmo através de um *binding* para a interface de operação do mesmo.

O servidor de aplicação implementa os objetos de serviço, os quais protege com ACL's SPKI a partir de um guardião. Para executar as delegações e negociações na propagação de permissões, o servidor também pode fazer uso de um agente similar ao que é definido no cliente. O servidor no processo de redução de certificados pode emitir certificados de autorização aos clientes que se apresentam com novas cadeias de delegação e/ou incluir as chaves públicas destes clientes nas ACL's do guardião.

3.2 GERENTE DE CERTIFICADOS

O gerente de certificados tem o objetivo de facilitar a interação entre o cliente e o servidor. Um gerente de certificados serve apenas ao grupo de principais de sua federação – as chaves públicas de seus integrantes formam um grupo SDSI. Um GC, não participando ativamente de nenhuma cadeia de autorização, não é caracterizado como uma chave pública (não é um principal); é principalmente um repositório de certificados públicos.

Para que um principal qualquer se filie a uma federação é necessário um endosso efetivado através da apresentação de um *threshold certificate* [Aura, 1998a]. A assinatura do *threshold certificate* depende de k-dentre-n membros da federação. O número *k* de membros necessários para endossar um pedido de filiação é definido por cada federação. Na filiação de um principal, o seu certificado de nome é incluído no repositório da federação. Este certificado de nome será mantido pelo gerente de certificados para auxiliar no processo de identificação de principais (seção 3.3). A todo novo membro incluído na federação é fornecido um "certificado de grupo" (certificado de nome expressando participação em um grupo SDSI) para fins de comprovação da filiação (*membership*).

O estabelecimento de associações entre federações (teias de federações) também é interpretado como admissão de novos membros nos "grupos SDSI" de cada federação envolvida. Só que neste caso, o novo membro (a outra federação) é tratado como um grupo definido e administrado em outro espaço de nomes – levando em conta a definição de grupos prevista no SDSI [Lampson e Rivest, 1996].

Portanto, ao gerente de certificados cabe a manutenção das informações referentes aos membros e associados de sua federação, removendo ou adicionando membros e associações com outras federações, sem promover conflito de interesses [Brewer e Nash, 1989]. Operações de armazenamento e de recuperação de certificados de nome e de autorização estão disponíveis aos membros da federação através de interfaces padronizadas oferecidas pelo GC – que na busca de cadeias de certificados se vale do algoritmo *Depth First Search* [Elien, 1998].

3.3 AUTENTICAÇÃO, AUTORIZAÇÃO E AUDITORIA NO MODELO

Na autenticação de principais SDSI / SPKI, a identificação não é feita através de nomes, mas de chaves públicas e o mecanismo de autenticação é a assinatura digital. Assim, para que a assinatura digital seja verificada no destino, a chave pública de um principal deve chegar de maneira segura até o destinatário. Como não há uma entidade distribuidora de chaves públicas certificadas no esquema SDSI / SPKI, as chaves públicas necessárias em uma autenticação são disponibilizadas através de uma cadeia de certificados de autorização.

A autenticação mútua no SDSI / SPKI é então conseguida com base em uma cadeia de autorização. O cliente ao fazer uma requisição a um servidor, deve assiná-la e enviá-la junto com a cadeia de autorização que lhe concede os privilégios de acesso necessários. Quando da chegada da requisição ao guardião a seqüência da cadeia de autorização será verificada. Se bem sucedida esta verificação, o guardião usa a última chave da cadeia de autorização (chave do cliente, constante no campo de sujeito) para verificar a assinatura digital da requisição. Com esta assinatura conferida fica confirmada a autenticidade do cliente.

Todo certificado de autorização tem no campo do emissor a chave pública do principal que assina o certificado. Logo, para autenticar um servidor que é sempre representado por uma chave pública iniciando uma cadeia de autorização (primeiro certificado da cadeia de autorização), o cliente precisará do certificado de nome do servidor, recuperado a partir de uma teia de federações. Então, o cliente usa a chave pública do certificado de nome para validar a assinatura do servidor na cadeia de autorização. Se todas as verificações citadas forem bem sucedidas, a identidade do servidor estará garantida.

Para efeito de auditoria, são usados também os registros de acessos (*log*s) das chaves públicas no servidor. Quando necessário é executada a busca do certificado de nome na teia de federações para identificar o principal correspondente à chave pública que efetivou o acesso.

Todos os mecanismos de autenticação e autorização citados nesta seção estão em conformidade com as especificações SDSI / SPKI.

4. FORMAÇÃO DE NOVAS CADEIAS DE AUTORIZAÇÃO NO MODELO PROPOSTO

Na literatura técnica científica são várias as experiências com a busca de cadeias de certificados SDSI / SPKI [Nikander e Viljanen, 1998][Aura, 1998b][Ajmani, 2000][Clark, 2001]. Porém, em todas estas abordagens, quando a cadeia de certificados não é encontrada, a busca termina com uma exceção (falha) e o cliente fica impossibilitado de efetuar o acesso pleiteado. Neste trabalho, através da noção de federações é criado um esquema que permite a um cliente localizar um certificado com a autorização desejada em uma teia de federações. Posteriormente, o cliente pode negociar com o possuidor do privilégio a concessão do mesmo para constituir uma cadeia de autorização que possibilite o acesso desejado.

Para ilustrar o processo de formação de novas cadeias, considera-se no exemplo da Figura 3 que inicialmente um certificado de autorização está armazenado no GC da federação *X*, após ter sido propagado do servidor *S* para o cliente *A* (*A* é membro da federação *X*). Na Figura 4, é descrita a seqüência necessária de mensagens trocadas quando da formação de uma cadeia inexistente entre o cliente *B* e o servidor *S*.

Assim, em algum momento o cliente *B*, membro da federação *Y*, faz um pedido de acesso ao servidor *S* (mensagem m_1, na Figura 4). O servidor *S*, por sua vez, envia um desafio em resposta a m_1. No desafio *S* informa a ACL protegendo o objeto solicitado e exige do cliente *B* a comprovação da autorização para o acesso requerido (mensagem m_2). Neste caso, as informações da ACL SDSI / SPKI são úteis para acelerar o processo de procura.

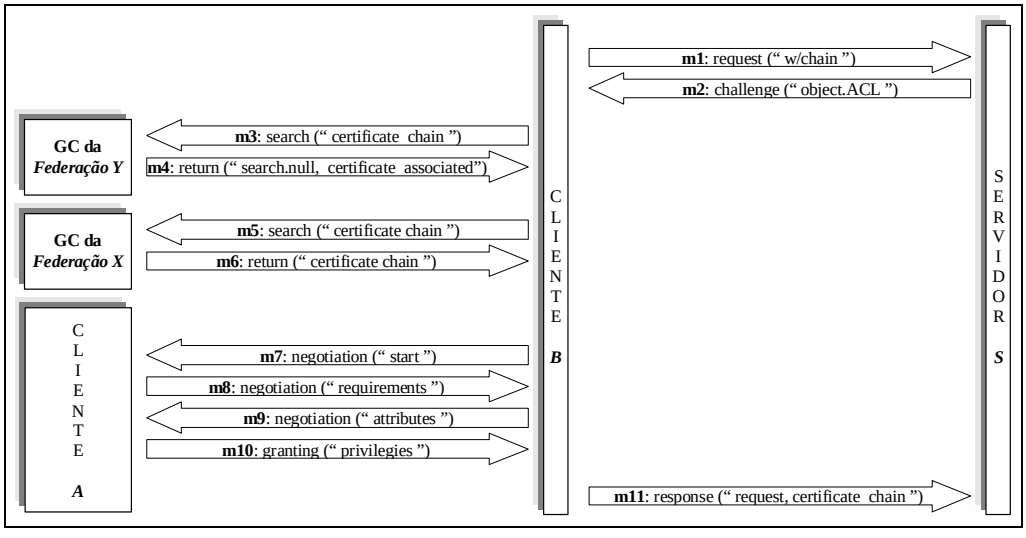

Figura 4 – Trocas de mensagens na formação de cadeia de autorização

Com as informações da ACL[2], o agente de *B* faz a busca local por uma cadeia de autorização que ligue o cliente *B* ao servidor *S* e permita o acesso desejado. A pesquisa deve retornar todas as cadeias de autorização que incluam a permissão necessária e tenham o servidor desejado como origem. Supondo que a busca local do exemplo resulte em insucesso, o agente de *B* recorre então ao GC da federação a que pertence (federação *Y*), encaminhando-lhe a pesquisa por certificados de autorização com os direitos requeridos para o acesso ao servidor *S* (mensagem m_3, na Figura 4).

Os critérios de busca considerados na pesquisa são as permissões requeridas e a chave pública de *S*. No caso considerado na Figura 4, esta pesquisa não resulta no retorno de nenhuma cadeia de autorização. Nesta situação, o GC da federação *Y* retorna ao cliente *B*, como resultado da pesquisa, os certificados de membro nas federações associadas (federação *X*, no exemplo da Figura 3) da teia de federações para que o mesmo possa prosseguir sua procura (mensagem m_4). De posse dos certificados de membro o cliente estende sua procura por GCs de outras federações da teia. A mensagem m_5 corresponde à consulta na federação *X* do exemplo considerado. Na mensagem m_6, o cliente *B* recebe como retorno do GC de *X* uma cadeia – o certificado de autorização com a permissão de acesso entre o cliente *A* e o servidor *S*.

O cliente *B* envia então ao possuidor dos direitos (cliente *A*) o pedido de delegação do direito (mensagem m_7, na Figura 4). A delegação de permissões a um solicitante pode ser concretizada de maneira simples – pelo fato do cliente e do detentor do direito compartilharem a mesma federação, por exemplo. Porém, dependendo da semântica da aplicação pode haver a necessidade de uma negociação dinâmica mais complexa. A Figura 4 ilustra esta última situação: o possuidor do direito requerido informa ao cliente *B* um conjunto de requisitos para a concessão pleiteada (mensagem m_8). O cliente reúne os requisitos exigidos e os envia ao cliente *A* (mensagem m_9). Uma vez que os requisitos da aplicação foram atendidos, o possuidor dos direitos emite o certificado delegando as permissões ao cliente *B* (mensagem m_{10}). Com esta última mensagem o processo de formação da cadeia é concluído e o cliente *B* pode responder ao *challenge* proposto pelo servidor *S* (mensagem m_{11} na Figura 4).

4.1 CASO: COMPRAS NA INTERNET

A seguir será descrito um cenário para ilustrar a utilização de teias de federações que sintetiza todo o esquema proposto. Este cenário construído sobre uma aplicação de compras via Internet envolve a localização e a negociação de privilégios de acesso. Porém, ressalta-se que esta aplicação poderia ser outra qualquer.

Para facilitar o entendimento do cenário, presume-se que uma administradora de cartões de crédito e um banco – com *CC* e *BK* como seus representantes, respectivamente – tenham algum tipo de parceria que lhes permita fácil compensação financeira. Com base nesta parceria, o representante *CC* delega ao representante *BK* o direito de "liberar compras" – no caso, o representante *BK* pode liberar compras quando o pagamento deveria

[2] No envio da ACL, o servidor passa ao solicitante do acesso informações de quais permissões são necessárias para o acesso desejado sobre o objeto e quais as chaves públicas possuem estas permissões.

ser feito com cartões de crédito da administradora a que o representante *CC* pertence. O representante *BK* ao receber o certificado de autorização, com *bit* de delegação ligado, o armazena junto ao GC da federação *FB* (federação que é membro).

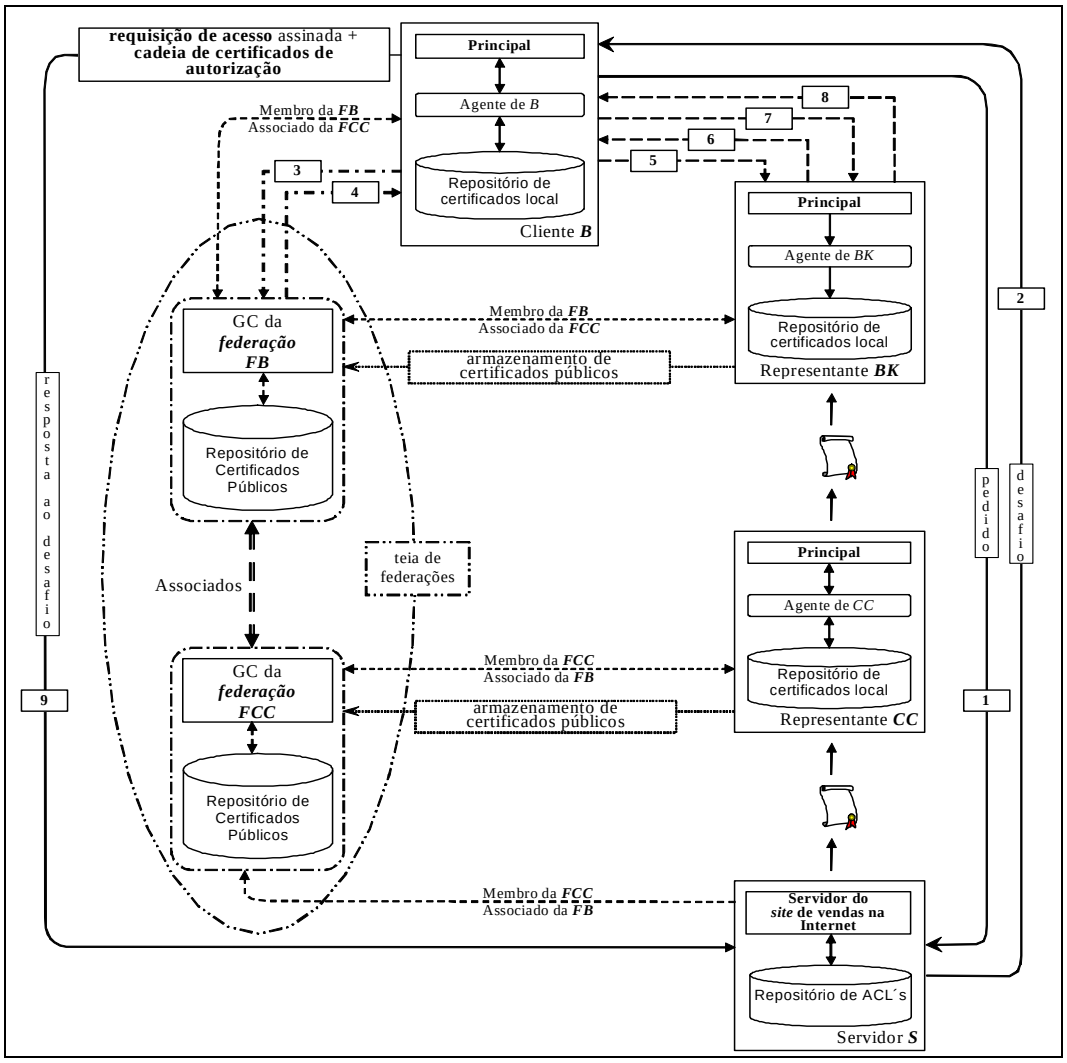

Figura 5 – Cenário para compras na Internet usando teia de federações

Na Tabela 2 são descritas as mensagens trocadas – identificadas numericamente na Figura 5 – entre as entidades da teia de federações que permitem a efetivação da transação de compra no site vendas na Internet.

Tabela 2 – Descrição das trocas envolvidas no cenário mostrado na Figura 5

Troca	Atividades
1	O cliente *B* carrega em seu browser as páginas do site de vendas na Internet implementadas no servidor *S*. Após navegar pelo site e selecionar itens, o cliente *B* escolhe a opção “ir ao caixa”.
2	O servidor *S* envia ao cliente uma mensagem desafio (*challenge*: ACL informando que o principal possuidor do privilégio “liberar compras” é o representante *CC* e solicitando ao cliente a comprovação de autorização emitida a partir de *CC*) junto com o bilhete de compras contendo o valor a pagar.

Troca	Atividades
3	Como o cliente *B* consultou seu repositório local e não encontrou nenhuma cadeia entre o representante *CC* e o cliente *B*, então, o cliente *B* envia uma mensagem de pesquisa de cadeia ao GC da federação *FB* (federação em que é membro) com os seguintes critérios de busca: chave pública do servidor *S* e operação "liberar compras" (direito requerido).
4	O GC da federação *FB* faz uma busca em seu repositório de certificados públicos e encontra a cadeia solicitada, então, retorna ao cliente a cadeia entre o servidor *S* e o representante *BK*.
5	O cliente *B* solicita a delegação do privilégio "liberar compras" no servidor *S* ao representante *BK*.
6	O representante *BK* informa ao cliente *B* que o requisito para a delegação do direito requerido é o pagamento do bilhete de compras, por exemplo, através de uma das opções: débito em conta corrente, boleto bancário,... .
7	O cliente *B* efetua o pagamento através de uma das opções oferecidas pelo representante *BK*.
8	O representante *BK*, então, delega o privilégio "liberar compras" no servidor *S* ao cliente *B*.
9	O cliente *B* envia a cadeia de autorização ao servidor *S* junto com o pedido (*response*) e o servidor finaliza o pedido de compra.

Para efeito de controle das liberações de compras feitas pelo representante *BK*, o representante *CC* recebe uma cópia dos bilhetes de compra pagos pelos clientes ao servidor do site de vendas na Internet e o elo então é fechado.

No cenário descrito acima, não existia uma cadeia de autorização entre o representante *CC* e o cliente *B*, porém, através da teia de federações foi possível criar de maneira dinâmica e automática a cadeia de autorização necessária para completar a operação de compra no site de vendas na Internet. Evidentemente, se a cadeia com a autorização requerida não fosse encontrada no GC da federação *FB* a pesquisa poderia continuar nas federações associadas até que a mesma fosse encontrada.

Para o cenário descrito na Figura 5, pode-se considerar que não é necessária uma entrada contendo o cliente *B* nas ACL's do servidor de destino para que o cliente esteja autorizado a ter acesso ao mesmo. Logo, não é necessário o tradicional cadastramento de usuários (clientes) no servidor para que os mesmos possam ter acesso a este servidor. Consequentemente, todos os dados cadastrais dos clientes estarão armazenados apenas nas instituições com as quais o cliente tem um relacionamento mais direto. No caso do exemplo acima, o cliente poderia efetuar uma compra na Internet e pagar pela compra como se fosse cliente de uma administradora de cartões de crédito, apenas sendo cliente de um banco que tem um relacionamento financeiro com esta administradora. Assim, não há números de cartão de crédito e nenhum outro dado pessoal do cliente circulando pela rede ou armazenado em base de dados de nenhum outro servidor que não seja o do banco onde o cliente tem conta corrente.

5. ASPECTOS DE IMPLEMENTAÇÃO DA ARQUITETURA

A infra-estrutura SDSI / SPKI e as políticas empregadas no modelo (descritas nas seções 3 e 4) são totalmente independentes da tecnologia usada. Neste sentido, as

ferramentas adotadas na composição da arquitetura do protótipo (Figura 6) foram fortemente influenciadas pelo uso do modelo na Internet – ambiente assumido como o contexto do trabalho.

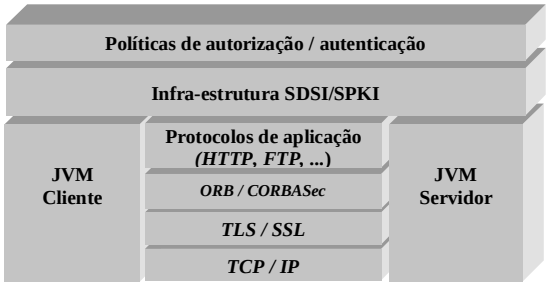

Figura 6 – Arquitetura do protótipo do modelo

A idéia fundamental do uso do CORBA *é* usufruir do suporte a objetos distribuídos, nos aspectos referentes à localização de objetos (resolução de nomes) e a segurança em invocações remotas. Como tecnologia de segurança usada nas comunicações remotas foi assumido o SSL (*Secure Socket Layer*). No caso, para que se estabeleça uma sessão entre cliente e servidor usando um canal seguro (com integridade e confidencialidade SSL), é necessária a autenticação mútua dos principais (cliente e servidor). Porém, o SPKI utiliza chaves como principais e não nomes. Assim foi necessário implementar funções para traduzir certificados de nome SDSI / SPKI para o SSL.

A integração do SDSI / SPKI ao suporte de objetos distribuídos foi implementada usando as especificações CORBASec [OMG, 2002], em nível de aplicação *(Security Level 2* do *CORBASec*). A Figura 7 ilustra esta integração. O *Security Level 2* é praticamente omisso na estruturação das funcionalidades de segurança em nível de aplicação. Porém, para obter benefícios do modelo de segurança do CORBA, um conjunto mínimo de objetos em nível de *ORB* foram mantidos, ou seja, estão sendo utilizados os objetos de sessão: *PrincipalAuthenticator, SecurityManager* e *Credentials.*

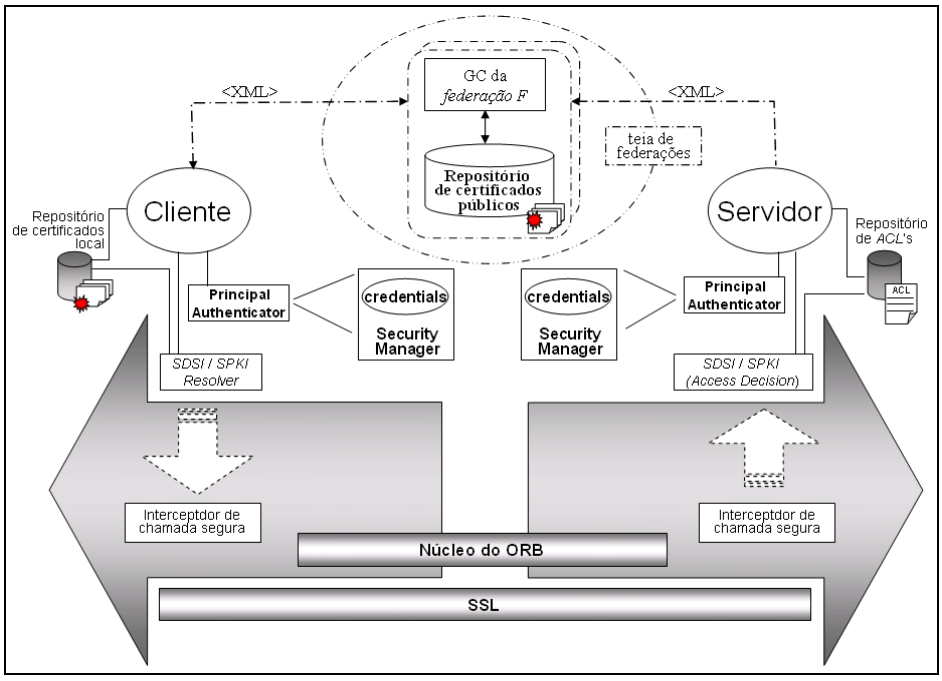

Figura 7 – Visão geral da implementação *CORBA-SPKI* do protótipo do modelo

A Figura 7 ilustra outros detalhes de implementação. O repositório de certificados públicos do gerente de certificados da federação é implementado com o Apache Xindice (armazena dados XML nativos) [Staken, 2002]. O gerente de certificados é projetado como um módulo de extensão do servidor Apache [Tchau, 2002]. Todas as trocas efetuadas entre os membros e o gerente de certificados estão sendo efetivadas usando XML. Os certificados SDSI / SPKI originalmente codificados em *S-expressions* estão sendo codificados no protótipo em XML [Terreros e Ribes, 2002] por razões de portabilidade e padronização. O objeto SDSI / SPKI *Resolver* da Figura 7 é uma implementação parcial do agente do cliente e executa principalmente as buscas da cadeia e atividades relacionadas com assinaturas digitais. Por fim, o monitor de referência (guardião) é implementado pelo objeto SDSI / SPKI *AccessDecision*. Ressalta-se que a integração de clientes e servidores ao ambiente do protótipo está sendo bastante facilitada devido ao emprego de *plugin*s e/ou *applet*s na construção da aplicação.

6 TRABALHOS RELACIONADOS

Em [Nikander e Viljanen, 1998], o DNS foi usado para armazenar e recuperar os certificados SDSI / SPKI. Na proposta foram utilizadas extensões acrescidas pela RFC 2065 ao DNS, para permitir o armazenamento de registros de certificados. Além disto, foi proposta uma infra-estrutura com entidades que armazenam os certificados de identificação e autorização no DNS e os algoritmos de busca que incluem a filtragem na recuperação dos registros de certificados pertencentes à cadeia de interesse.

Em [Aura, 1998b] é considerado que a rede formada pela propagação de certificados de autorização SDSI / SPKI pode ser interpretada como um grafo direcionado. Além disto, é assumido ainda, que em ambientes tipicamente organizacionais tal rede tem a forma de uma ampulheta. Isto devido à constatação da ocorrência de um número maior de chaves de servidores e de clientes, do que de chaves intermediárias entre ambos. Então o autor, a partir destas premissas, utiliza os algoritmos DFS forward, DFS backward e uma combinação de ambos para fazer buscas rápidas numa base com um único intermediário.

Em [Ajmani, 2000] é relatada a experiência de implementação da busca distribuída dos algoritmos propostos em [Aura, 1998b] – mais especificamente, uma proposta para que o algoritmo DFS forward apresentasse melhores resultados.

Em linhas gerais pode-se observar nos trabalhos descritos em [Nikander e Viljanen, 1998] e [Aura, 1998b] que estes foram concebidos para versões preliminares de SDSI / SPKI, onde alguns aspectos do modelo ainda não estavam bem resolvidos. Portanto, algumas premissas assumidas na época, atualmente, não estão mais em consonância com as recomendações da RFC 2693. Já no trabalho de [Ajmani, 2000] praticamente se tem uma implementação resolvendo um problema de [Aura, 1998b]. Porém, em termos de arquitetura, nestas experiências boas idéias foram apresentadas.

Em [Li, 2000] é mostrado que os nomes locais SDSI / SPKI podem ser interpretados como grupos (conjunto de principais) distribuídos para a “resolução” de nomes. Assim, o autor desenvolve algoritmos baseados na programação em lógica para sustentar sua argüição e para justificar que estes são mais eficientes na busca das cadeias que as implementações convencionais. No caso, como o objetivo principal foi criar algoritmos de busca voltados para a programação em lógica, nenhuma arquitetura foi proposta. Porém, a interpretação de nomes locais como grupos distribuídos trouxe uma contribuição bastante importante.

No trabalho de [Clarke, 2001], os algoritmos de busca sugeridos e os demais aspectos considerados são, na verdade, bons refinamentos das recomendações feitas na RFC 2693. Além disto, um relato de implementação da versão atual de SPKI bastante rico em conteúdo é apresentado, mas não se tem uma proposta de arquitetura para sistemas distribuídos, por exemplo.

7. CONCLUSÃO

Como o gerente de certificados não figura nas cadeias de autorização como chave nas delegações sucessivas, o modelo proposto é totalmente descentralizado. Assim, o gerente não centraliza ou torna hierárquicas as relações de confiança entre cliente e servidor e nem assume o papel de ponto crítico em relação à falhas e a vulnerabilidades ou ao desempenho do sistema.

Em aplicações da Internet, o esquema proposto permite uma flexibilidade maior em aspectos referentes à legislação do que a infra-estrutura X.509, por exemplo. Considerando que um cliente geralmente negociará a concessão de privilégios com um principal de seus domínios (da mesma localidade – país, por exemplo) e que este principal por sua vez poderá estar inserido em domínios remotos. Então, haverá vínculos fortes entre o cliente e o principal local e entre este principal local e os principais dos domínios remotos, de modo que arbitrariamente, se definem contextos de abrangência compatíveis com os universos onde cada principal atua.

A adoção da teia de federações isenta o servidor do gerenciamento de usuários e também isenta o cliente da necessidade da tradicional criação de uma conta (*account*) para ter acesso a um servidor – mesmo num contexto global.

O modelo proposto apresenta um suporte à gerência de certificados que permite a criação de novas cadeias de autorização que não é verificado em nenhum dos trabalhos relacionados. O esquema proposto é bastante flexível e automático, mesmo considerando que em alguns casos o número de trocas necessárias para criar a cadeia possa ser expressivo.

AGRADECIMENTOS

Os autores agradecem ao apoio financeiro do CNPq, no programa de conteúdos digitais, ao projeto de processo CNPq nº 552175/2001-3. À Priscila pelas revisões do texto.

REFERÊNCIAS BIBLIOGRÁFICAS

AJMANI, Sameer (2000). *A trusted Execution Platform for Multiparty Computation.* Master thesis, Dep. of Electrical Engineering and Computer Science, MIT.

AURA, Thomas (1998a). *On the Structure of Delegation Networks*. In: proceedings of 11th IEEE Computer Security Foundations Workshop.

AURA, Thomas (1998b). *Fast Access Control Decisions from Delegation Certificate Databases*. In: proceedings of 3th Australasian Conference on Information Security and Privacy.

BLAZE, M., FEIGENBAUM, J., LACY, J. (1996). *Decentralized Trust Management*. In: Proceedings of the 17th IEEE Symposium on Security and Privacy.

BLAZE, M., FEIGENBAUM, J., LACY, J. (1999). *The KeyNote Trust Management System, Version 2*. IETF *RFC2704*.

BREWER, D., NASH, M.(1989). *The chinese Wall Security Policy*. In: Proceedings of IEEE Symposium on Security and Privacy.

CLARKE, Dwaine E. (2001). *SPKI/SDSI HTTP Server Certificate Chain Discovery in SPKI/SDSI*. Master dissertation, Dep. Electrical Engineering and Computer Science of MIT.

ELIEN, Jean-Emile (1998). *Certificate Discovery Using SPKI/SDSI Certificates*. Master dissertation, Dep. Electrical Engineering and Computer Science of MIT.

ELLISON, C., FRANTZ, B., LAMPSON, B., RIVEST, R., THOMAS, B., YLONEN, T. (1999). *SPKI Certificate Theory*. IETF *RFC2693*.

GARFINKEL, Simson (1995). *PGP:Pretty Good Privacy*. O'Reilly & Associates, Inc.

GASSER, M., McDERMOTT, E. (1990). *An Architecture for Practical Delegation in a Distributed System*. In: proceedings of the IEEE Symposium on Security and Privacy.

OMG – Object Management Group (2002). *Security Service specification, v1.8*. [online] Disponível em http://www.omg.org/cgi-bin/doc?formal/02-03-11.pdf. Acesso em: janeiro de 2003.

HORST, F. W., LISCHKA, M. (2001). *Modular Authorization*. In: Proceedings of the Sixth ACM Symposium on Access control models and technologies.

LAMPSON, B., RIVEST, R. L. (1996). *A Simple Distributed Security Infrastructure*. [online] Disponível em http://theory.lcs.mit.edu/~cis/sdsi.html. Acesso em janeiro de 2003.

LI, Ninghui (2000). *Local Names in SPKI/SDSI*. In: proceedings of the IEEE Computer Security Foundations Workshop.

NIKANDER, P., VILJANEN, L. (1998). *Storing and Retrieving Internet Certificates*. In: 3th Nordic Workshop on Secure IT Systems.

STAKEN, K. (2002). *Xindice Developers Guide 0.7.1*. Disponível em http://xml.apache.org/xindice/guide-developer.html. Acesso em janeiro de 2003.

TERREROS, Xavier Orri Sainz de los, RIBES, Joan-Maria Mas (2002). *SPKI-XML Certificate Structure*. [online] Disponível em http://www.oasis-open.org/cover/xml-spki.html. Acesso em janeiro de 2003.

THAU, Rob. (2002). *Design Considerations for the Apache API*. [online] Disponível em http://modules.apache.org/reference. Acesso em janeiro de 2003.