UM MODELO DE AUTORIZAÇÃO E AUTENTICAÇÃO BASEADO EM REDES DE CONFIANÇA PARA SISTEMAS DISTRIBUÍDOS DE LARGA ESCALA

Altair Olivo Santin, Joni da Silva Fraga, Emerson Ribeiro de Mello, Frank Siqueira(*)
Departamento de Automação e Sistemas / LCMI e (*)Departamento de Informática e Estatística
Universidade Federal de Santa Catarina
Florianópolis – SC – CEP: 88040-900 – Brasil
{santin, fraga, emerson}@das.ufsc.br, frank@inf.ufsc.br

RESUMO

Este trabalho apresenta um modelo de autorização e autenticação que visa minimizar, principalmente, as dificuldades de escalabilidade e flexibilidade dos sistemas clássicos, em ambiente de larga escala como a Internet. O modelo se baseia em redes de confiança construídas a partir da delegação de privilégios de acesso, codificados em certificados de autorização e armazenados num repositório local (gerente de certificados) que representa o agrupamento de principais com um mesmo propósito – a federação. Através da construção de relações de confiança entre federações é constituída uma teia de federações que dá um sentido global a tais certificados. A principal função da teia é auxiliar o cliente na busca dos privilégios de acesso que o ligam a um servidor. Após encontrá-los, o cliente poderá negociar a concessão de tais privilégios de acesso junto ao detentor dos mesmos.

ABSTRACT

This work presents an authorization and authentication model, which main purpose is minimizing the scalability and flexibility difficulties of classic systems in large-scale environment – as Internet, for example. The model is based upon trust chains built from access delegation privileges, coded in authorization certificates and stored in a local repository (certificate manager), that represents a grouping of principals (clients and servers) all with a same purpose – the federation. By the trust relationships construction among the federations is constituted a web of federations. These federations provide a global sense to such certificates. The mainly federations web function is helping the client in the access privileges searches that can link one to the server. After the access privileges locating, the client can start the granting negotiation from the holder of these access privileges .

1 INTRODUÇÃO

A abordagem clássica, usada na autenticação e autorização em sistemas distribuídos, define uma autenticação centralizada, baseada em domínios de nomes e precedendo a autorização que geralmente tem os seus controles distribuídos. Este modelo, empregado em redes corporativas, parece não ser muito adequado quando o ambiente é a rede mundial, onde o cliente (usuário) muitas vezes não é conhecido de antemão.

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 a falhas e a vulnerabilidades, pode impor sérias restrições ao desempenho e à escalabilidade do sistema em ambiente de larga escala (Horst e Lischla, 2001).

As abordagens que se propõem a atender aos requisitos de escalabilidade e flexibilidade em tal ambiente, estão baseadas em infra-estruturas de chaves públicas ou *PKI*s (*Public Key Infrastructures*). Atualmente, a mais comum destas infra-estruturas é o *X.509*, que define um modelo hierárquico de confiança baseado na nomenclatura *X.500*. As comunidades *X.509* devem ser construídas com base na confiança das chaves privadas das *Certification Authorities* (*CA*'s) formando a hierarquia global. Cada *CA* controla sua chave privada que é usada na assinatura digital dos certificados que emite. Estes certificados vinculam nomes de validade global as respectivas chaves públicas. O modelo, essencialmente, baseia-se em certificados formando cadeias (*chain*s) de autenticação a partir de uma *CA* raiz (*root*) até um usuário (principal).

Quando são consideradas aplicações na Internet, a autorização e a autenticação devem evoluir tomando como base modelos, onde as relações de confiança possam ser estabelecidas de maneira distribuída, escalável e flexível.

Na infra-estrutura *X.509*, é possível construir relações de confiança através de certificação cruzada entre duas *CA*'s. O objetivo é dispensar a necessidade de percorrer toda a estrutura hierárquica para ir de uma *CA* até outra. Porém, em tal modelo, este tipo de alternativa de fluxo só é permitido entre *CA*'s. Evidentemente, isto não descaracteriza a *CA* como entidade centralizadora da certificação (autenticação) em seu domínio de política, nem a rigidez e a complexidade da estrutura hierárquica imposta pela *PKI X.509*.

Para o cenário de aplicações como as da rede mundial, seria mais adequado um modelo onde as relações de confiança pudessem ser estabelecidas sem nenhuma entidade centralizadora. Isto poderia ser conseguido construindo-se relacionamentos do tipo *peer-to-peer*, por exemplo, ou através de cadeias de confiança, onde cada entidade da cadeia

(caminho) se comportaria como se fosse uma *CA* do modelo *X.509* (Hastings e Polk, 2000).

A infra-estrutura *Pretty Good Privacy* (*PGP*) foi desenvolvida por *Phil Zimmermann* em 1991, para cifragem e autenticação de arquivos e correio eletrônico (Garfinkel, 1995). No *PGP* é utilizada criptografia de chave pública e uma estrutura para gerenciamento e certificação das chaves baseada na chamada teia de confiança (*web of trust*). A teia de confiança não se baseia em hierarquias como o *X.509*. Ao invés disto, os usuários *PGP* podem construir caminhos de confiança de forma arbitrária, através das comunidades *PGP* ao redor do mundo.

Um certificado de nome no *PGP* também liga um nome global a uma chave pública e pode ser emitido por qualquer principal. Neste caso, os certificados não são emitidos por *CA*'s com responsabilidades legais, mas por usuários comuns. O modelo de teia de confiança parece funcionar bem em comunidades que interagem de maneira pouco freqüente. Porém, este modelo pode mostrar-se debilitado na tomada de decisões de confiança, pois requer múltiplas assinaturas dando credibilidade ao certificado em avaliação.

O controle de acesso é tradicionalmente baseado na confrontação da identidade autenticada do cliente, que deseja acessar um recurso, com as *ACL's* (lista de controle de acesso) armazenadas na memória local do *guardião* (núcleo de segurança) do recurso sendo acessado. Para a Internet, quando usadas as infra-estruturas citadas acima, a autenticidade da identidade dos clientes baseia-se em assinaturas digitais e certificados de nome. Esquemas de autorização montados a partir destes certificados de nome, como os empregados no *X.509* ou *PGP*, são comumente denominados de "*orientados a nomes*".

Os autores (Blaze, Feigenbaum e Lacy, 1996) argumentam que o uso de infra-estruturas de chave pública baseadas em nomes e *ACL*'s são uma solução inadequada para o controle de acesso em sistemas como a rede mundial. Tal argüição baseia-se principalmente na percepção das dificuldades de escalabilidade e da falta de flexibilidade destas infra-estruturas de nomes globais. Além disto, *CA*'s agindo como entidades legais – com base nas hierarquias de confiança – em diferentes países com soberania e legislação própria, são apontadas como uma das principais limitações do esquema *X.509*.

Como uma alternativa, no trabalho de (Blaze, Feigenbaum e Lacy, 1996) é introduzido o conceito de gerência de confiança (*trust management*) compreendendo um conjunto de mecanismos unificados para especificar e validar políticas de autorização, credenciais e seus relacionamentos. Estes mecanismos de gerência de confiança são geralmente denominados de "*orientados a chaves*", por ligarem chaves públicas diretamente a autorizações.

Neste trabalho será mostrado o uso de cadeias de confiança na efetivação de um modelo de autenticação e autorização para sistemas distribuídos de larga escala. O modelo de confiança proposto se baseia no estabelecimento arbitrário de relações de confiança em âmbito local (federação), com o objetivo de figurar numa composição de abrangência global (teia de federações).

O restante do texto está organizado da seguinte maneira. A seção 2, revisa a gerência de confiança. A seção 3, introduz o modelo de autenticação e autorização proposto. A seção 4, aborda considerações sobre os aspectos de implementação do modelo proposto. A seção 5, expõe os trabalhos correlatos, e a seção 6, apresenta conclusões acerca deste trabalho

2 GERÊNCIA DE CONFIANÇA

A gerência de confiança unifica a noção de política de segurança, credenciais, controle de acesso e autorização, em duas abordagens distintas. Em uma, a gerência de confiança é efetivada usando uma linguagem na descrição das políticas e das credenciais, e um motor-lógico (*engine*) define o comportamento do módulo de checagem de conformidade (*compliance checker*). Na outra abordagem, uma estrutura de dados padronizada é utilizada para descrever os atributos de segurança que definem as políticas e os certificados no sistema. Em ambas as abordagens, a checagem de conformidade indica se os certificados (credenciais) apresentados por um principal estão em conformidade com a política local, especificada pelo guardião.

A seguir são mostrados os principais sistemas baseados em gerência de confiança.

2.1 PolicyMaker

No ambiente do *PolicyMaker* (Blaze, Feigenbaum e Lacy, 1996) o objetivo é a autorização, expressa através de asserções. Uma asserção é um par (*f*, *s*), onde *s* representa a autoridade origem (emissor) e *f* descreve a autorização concedida e o seu beneficiário. Na especificação das asserções, uma linguagem é usada como um meio correto e inequívoco de expressão e de interpretação das mesmas. Esta linguagem comum se faz necessária, porque um ambiente local pode importar credenciais de diversas origens nas verificações de conformidade. O *PolicyMaker* não define uma linguagem única para a especificação de asserções, porque sua principal preocupação é com o formalismo da

especificação. Assim, a checagem de conformidade é independente das linguagens de especificação de políticas e de credenciais. Além disto, boa parte das atividades da gerência de confiança como a verificação de assinaturas, por exemplo, é atribuída à aplicação que, assim, pode escolher livremente a tecnologia de segurança a ser utilizada.

Basicamente, o módulo de checagem de conformidade toma como entrada um conjunto de asserções de credencial (C), requisições (r) e asserções de política (P) e gera como saída um parecer refletindo a conformidade ou não de *C* e *r* com *P*. As credenciais em *C* devem formar cadeias de delegação que concedem ao requerente as permissões para o acesso desejado. Não é função do módulo de checagem descobrir credenciais ausentes em *C* e buscá-las. Tal módulo apenas decide quais asserções devem ser avaliadas e em qual ordem, para gerar todos os registros parciais de aceitação. Para isto uma estratégia de buscas repetidas é usada na geração dos registros parciais. No fim do processo, estes registros são ordenados para que o módulo possa concluir se os mesmos representam um conjunto substancial de provas de conformidade para a requisição em questão ou não.

2.2 Keynote

O *KeyNote* (Blaze e outros, 1999) foi desenvolvido sob os mesmos princípios do *PolicyMaker*. Porém, no *Keynote* uma linguagem específica para definição de asserções foi adotada para aumentar sua eficiência e para facilitar a interoperabilidade na escrita de políticas e de credenciais. Isto permitiu repassar mais atribuições para o módulo de gerência de confiança – é o caso da checagem da assinatura digital. Além disto, na busca das provas de conformidade foi adotado o algoritmo *DFS* (*Depth First Search*), que é executado no conjunto de credenciais do requerente. Assim, a estrutura resultante pôde ser tratada como um grafo direcionado durante a checagem, o que mostrou-se mais eficiente que a estratégia de buscas repetidas do *PolicyMaker* – que gerava registros parciais utilizados na construção das provas de conformidade.

2.3 SDSI / SPKI

A infra-estrutura *Simple Distributed Security Infrastructure / Simple Public Key Infrastructure* (*SDSI / SPKI*) foi motivada pela percepção da complexidade do esquema global de nomeação *X.509*. O *SPKI* (Ellison e outros, 1999) e o *SDSI* (Lampson and Rivest, 1996) são duas propostas de propósitos complementares. O *SDSI*, projetado no *MIT* por *Ronald Rivest* e *Butler Lampson*, é uma infra-estrutura de segurança com o objetivo principal de facilitar a construção de sistemas distribuídos seguros e escaláveis. O *SPKI* é o resultado dos esforços conduzidos por *Carl Ellison* no projeto de um modelo de autorização simples, implementável e bem definido. A combinação das duas propostas forma uma base para a autenticação e autorização em aplicações distribuídas. Nesta infra-estrutura os espaços de nomes de principais[1] são locais e o modelo, baseado em cadeias de confiança, é simples e flexível.

Na atual versão do *SDSI / SPKI* há dois tipos distintos de certificados, um para nome e outro para autorização. Os certificados de nome (Tabela 2.1) ligam nomes a chaves públicas, ou ainda, a outros nomes. O sistema de nomeação é herdado do *SDSI* que induz ao uso de nomes locais 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 – limitados ao espaço de nomes do emissor do certificado. A combinação da chave pública do emissor do certificado com um nome local representa um identificador global único em *SDSI / SPKI*, sendo que o emissor de um certificado é sempre uma chave pública.

Tabela 2.1: Certificado de Nome *SPKI*

Campos	**Descrição**
Emissor (*Issuer*)	Chave pública da entidade (chave emissora) que está definindo o "Nome" em seu espaço de nomes local.
Nome (*Name*)	Nome local que está sendo atribuído ao sujeito.
Sujeito (*Subject*)	Uma chave pública ou um nome definido em outro espaço de nomes que será redefinido (referenciado) no espaço de nomes local ao emissor.
Validade (*Validity dates*)	Especificação do período de validade do certificado – em formato 'data-hora'.

No *SDSI / SPKI* é usado um modelo igualitário: os principais são chaves públicas que podem assinar e divulgar certificados, como uma *CA* do *X.509*. Assim, qualquer principal pode criar seu par de chaves (privada e pública) e então, associar um nome à chave pública do par – que pode ser 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 uma cadeia de nomes através do encadeamento de referências (*linked names*).

Os certificados de autorização (Tabela 2.2) *SDSI / SPKI* ligam autorizações a um nome, a um "grupo especial" de principais (*threshold subjects*) ou a uma chave. Através destes certificados, o emissor delega permissões de acesso a outros principais no sistema.

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

Na infra-estrutura *SDSI / SPKI* os certificados de autorização são construídos a partir das *ACL's* do guardião. Na verdade, no que se refere a certificados e *ACL's*, o *SDSI / SPKI* define um formato único de representação (Figura 3.3), facilitando as atribuições e checagens de autorização. O conteúdo do certificado pode ser o mesmo da *ACL*, porém, ao certificado é acrescido o campo do emissor assinando o certificado – a *ACL* não possui este campo porque é local ao guardião do serviço.

Tabela 2.2 – Certificado de autorização *SPKI*

CAMPOS	*Descrição*
Emissor (*Issuer*)	Chave pública do emissor do certificado.
Sujeito (*Subject*)	Chave pública (ou *hash* dessa) ou nome identificando o principal que receberá a autorização.
Delegação (*Delegation*)	Valor lógico (*True / False*), indicando se o sujeito pode (*True*) ou não (*False*) propagar a autorização que lhe foi delegada pelo emissor.
Autorização (*Authorization*)	Contém as permissões concedidas pelo emissor – representadas como *S-expression*.
Validade (*Validity dates*)	Especificação do período de validade do certificado – em formato 'data-hora'.

Em *SDSI / SPKI* os nomes de principais são utilizados apenas para facilitar a memorização da identificação dos mesmos (chaves públicas). Quando uma decisão de acesso vai ser tomada, os nomes precisam ser "resolvidos" – a cadeia de nomes ligados deve ser percorrida até chegar a chave pública do principal em questão. O processo de recuperação da cadeia de nomes para alcançar o certificado de nome equivalente é chamado de "*redução da cadeia de nomes*". Analogamente, os direitos concedidos através das delegações sucessivas, devem ser "*reduzidos* / resumidos" em um certificado único contendo os direitos propagados até um sujeito.

Considerando que os certificados (*E1*, *S1*, *D1*, *A1*, *V1*) e (*E2*, *S2*, *D2*, *A2*, *V2*) seguem o formato da Tabela 2.2. Então, estes certificados formam uma cadeia de autorização se as seguintes condições forem satisfeitas: *S1 = E2* e *D1 = 'True'*. Caso afirmativo, a redução da cadeia de autorização resultará um certificado com o seguinte formato: [*E1*, *S2*, *D2*, *intersect*(*A1*, *A2*), *intersect*(*V1*, *V2*)]. A função *intersect* no campo de autorização representa a interseção (*string* a *string*) de *A1* com *A2*. Já a função *intersect* (*V1*, *V2*), no campo de validade, se refere a interseção dos períodos de validade dos certificados em questão. Se não houver nenhuma interseção, a função de redução deve retornar uma exceção.

A especificação *SDSI / SPKI* prevê também recursos para tolerância a intrusões e a faltas, efetivados através dos *threshold subjects*, que são aplicáveis apenas aos certificados de autorização. Certificados com *threshold subjects* determinam que um número mínimo de sujeitos, *k* dentre *n*, devem autorizar a delegação de uma permissão para que a mesma se torne efetiva.

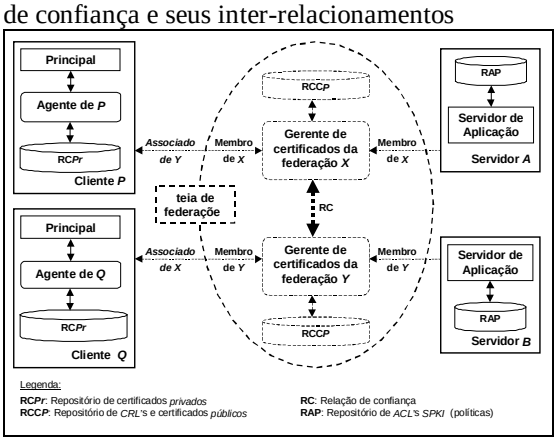
3 MODELO DE AUTENTICAÇÃO E AUTORIZAÇÃO PROPOSTO

Nesta seção são considerados aspectos referentes ao modelo de confiança, autorização, autenticação, auditoria e formação de novas cadeias de autorização no modelo proposto.

3.1 Modelo de Confiança

O modelo de confiança proposto está fundamentado no conceito de "*federação*" que enfatiza o agrupamento de principais, com interesses afins, definindo relações de confiança entre seus membros. 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 do compartilhamento dos certificados de seus membros.

Figura 3.1 – Visão geral das entidades do modelo de confiança e seus inter-relacionamentos



O modelo de confiança está centrado em três entidades: cliente, servidor (de aplicação) e gerente de certificados da federação (Figura 3.1). O gerente de certificados (entidade lógica para o processo de autorização, pois não participa da cadeia de autorização) tem o objetivo de facilitar a interação entre o cliente e o servidor. Este gerente oferece recursos para o armazenamento e a recuperação dos certificados de nome e de autorização, que os clientes necessitam para efetivação de acessos aos servidores.

Um gerente de certificados serve a apenas um grupo de principais – sua federação. Por exemplo, o gerente de certificados da federação dos clientes de um banco, armazena e recupera certificados de nome e de autorização apenas para os principais congregados pela federação do banco em questão. Todos os membros da federação podem compartilhar o acesso ao repositório de certificados, implementado pelo gerente de certificados da federação.

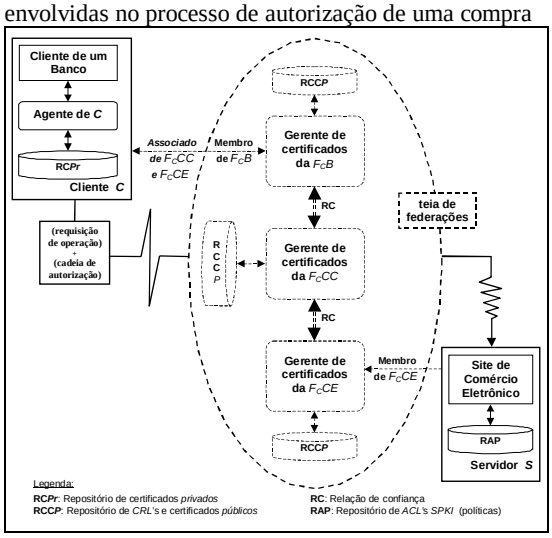
Os clientes e servidores podem filiar-se a várias federações, em tantas quantas forem

aceitos. Assim, por exemplo, pode-se imaginar um cenário (Figura 3.1), onde um cliente (*C*) que é membro da federação dos clientes de um banco (F_CB), se torna membro da federação dos clientes de uma administradora de cartões de crédito (F_CCC). Este mesmo cliente, pode filiar-se também a federação dos clientes de um *site* de comércio eletrônico (F_CCE). Paralelamente, o servidor (*S*) do *site* de comércio eletrônico (que é filiado a F_CCE), pode filiar-se a F_CCC. Assim, o cliente *C* poderia fazer compras no *site* de comércio eletrônico e pagar com um cartão de crédito oriundo de F_CCC, porque ambos (*C* e *S*) compartilham relações de confiança com as mesmas federações (F_CCE e F_CCC).

As filiações dos clientes a várias federações permite-lhes ter o acesso facilitado principalmente aos certificados de autorização dos membros da federação. Porém, o número de filiações necessárias para ter um determinado nível de presença na rede mundial caracteriza-se também como um problema de escalabilidade.

Os requisitos de escalabilidade deste modelo induzem a algum tipo de formação que resulte em uma espécie de teia de federações em âmbito global. Então, parece razoável que os gerentes de certificados associem-se uns aos outros – a aqueles que, por afinidade, melhor representem as necessidades de seus membros – através de relações de confiança para formar teias de federações com alcance global. Assim, os clientes e os servidores estariam eximidos da necessidade de filiar-se a *n* federações para ter escopo global.

Figura 3.2 – Exemplo de uma teia de federações envolvidas no processo de autorização de uma compra



Considerando-se então o caso anterior (Figura 3.1) no cenário de uma teia global de federações, se poderia vislumbrar uma situação como a mostrada na Figura 3.2. Neste caso, o cliente de um banco (*C*) estaria filiado a sua F_CB e o servidor (*S*) do *site* de comércio eletrônico estaria filiado a sua F_CCE. O gerente de certificados da F_CB e o gerente de certificados da F_CCC estariam associados entre si, sendo que o gerente de certificados da F_CCC estaria também em associação mútua com o gerente de certificados da F_CCE. Consequentemente, o cliente *C* poderia fazer compras no *site* de comércio eletrônico e pagar com um cartão de crédito oriundo da F_CCC, apenas pertencendo a F_CB, porque há uma relação de confiança entre ambos (*C* e *S*), através da teia de federações.

Pode-se observar que no modelo de confiança proposto não há nenhum tipo de centralização ou hierarquização. A teia de federações é formada arbitrariamente e desempenha funções de apoio (suporte) ao processo de autorização e autenticação, como será mostrado nos tópicos a seguir.

É oportuno salientar que a efetivação do modelo poderia ser alcançada com a utilização do *PolicyMaker* / *KeyNote*. Porém, a adoção do *SDSI / SPKI* no nosso modelo se deu em função do seu grau mais avançado de desenvolvimento, resultante das várias contribuições que este já recebeu dos pesquisadores da área.

3.1.1 Nome de Principais

Seguindo as especificações *SDSI / SPKI*, no modelo, os nomes de principais são locais. A publicação dos mesmos é feita através de certificados de nome assinados, que qualquer principal poderá emitir. Os nomes locais são concatenados a chave pública do principal, dono do espaço de nomes, para ter escopo global e são propagados através de cadeias de nomes ligados.

A noção de grupo também é adotada no modelo e corresponde a um conjunto de principais interpretados como um único nome local no espaço de nomes do principal que definiu o grupo.

3.1.2 Certificados de Autorização

Os certificados de autorização *SDSI / SPKI*, utilizados no modelo, são considerados em duas circunstâncias distintas. Na primeira, quando o bit de delegação estiver desligado (delegação não permitida), os atributos de privilégio não podem ser propagados. Neste caso, o principal deve guardar este certificado como sendo um certificado de autorização "*privado*", do qual só o mesmo pode fazer uso. Na segunda circunstância, quando o bit de delegação estiver ligado (delegação permitida), o sujeito pode fazer dos atributos de privilégio o que bem entender ou o que lhe for solicitado. Entre as atitudes que o sujeito pode tomar em relação aos atributos recebidos estão: guardá-los para seu uso particular (privado) ou repassá-los a terceiros na

íntegra ou num subconjunto, quando isto for necessário.

É importante observar que uma vez concedido um privilégio (delegável ou não), este é irrevogável, estando limitado apenas as restrições de uso temporal, especificadas no campo de validade do certificado. Assim, é possível considerar que os certificados de autorização *SPKI* podem representar não apenas simples atributos de privilégio, mas confiança irrestrita do emissor no sujeito – para administrar aqueles atributos de privilégio (Figura 3.3) durante o período de vigência imposto pelas restrições de validade. Isto permite uma descentralização da administração da concessão de direitos de acesso, que se bem explorada, propicia uma flexibilidade bastante considerável deste modelo sobre os convencionais.

3.1.3 Gerente de Certificados

Ao gerente de certificados cabe a manutenção das informações referentes aos membros e associados de sua federação – remoção ou adição de membros e associação a outras federações, sem promover conflito de interesses (Brewer e Nash, 1989). Ao gerente de certificados cabe também o armazenamento e recuperação dos certificados de nome e de autorização para os membros da federação. O armazenamento é feito em repositório(s) apropriado(s) do gerente e a recuperação é processada através de algoritmos de busca de cadeias (*Depth First Search*, por exemplo).

Os certificados de autorização que um membro armazena no gerente de certificados tornam-se de conhecimento dos demais membros e associados da federação ("*públicos*"), e só podem ser do tipo delegáveis. Estes certificados estarão à disposição dos membros e associados para possíveis pesquisas (buscas) da cadeia de autorização (Figura 3.3).

Considerando-se então um cliente (membro ou associado), desejando acessar um servidor, sem a autorização necessária. Tal cliente poderá fazer uma busca na teia de federações para identificar algum possuidor do direito requerido. Após localizar o direito, o cliente poderá iniciar uma negociação com o detentor do mesmo, objetivando obter sua concessão (veja seção 3.5).

3.2 Autenticação

Na autenticação de principais 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 (neste caso, identificador do principal e elemento de verificação da assinatura digital) deve chegar de maneira segura até o destinatário. Como não há um servidor para "registrar" a chave pública do principal, essa chave chegará ao destino por delegação – através das entidades que formam a cadeia de confiança na propagação dos certificados de autorização (Figura 3.3).

Em sistemas baseados em redes de confiança, todo o certificado de autorização tem no campo do emissor a chave pública do principal que assina o certificado (Tabela 2.2). Logo, todos os certificados de autorização podem ter sua autenticidade facilmente verificada. Mesmo o cliente que é o sujeito (última chave) de uma cadeia de autorização pode ter sua autenticidade facilmente verificada. Isto porque, ao fazer uma requisição a um servidor, o cliente deverá assiná-la e enviá-la junto com a cadeia de autorização que lhe concede os privilégios de acesso necessários. Então, quando o pedido chegar ao guardião, a cadeia de autorização será verificada (Tabela 3.2). Após uma verificação bem sucedida, 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. Se a assinatura confere, a autenticidade da requisição é confirmada, em caso contrário não.

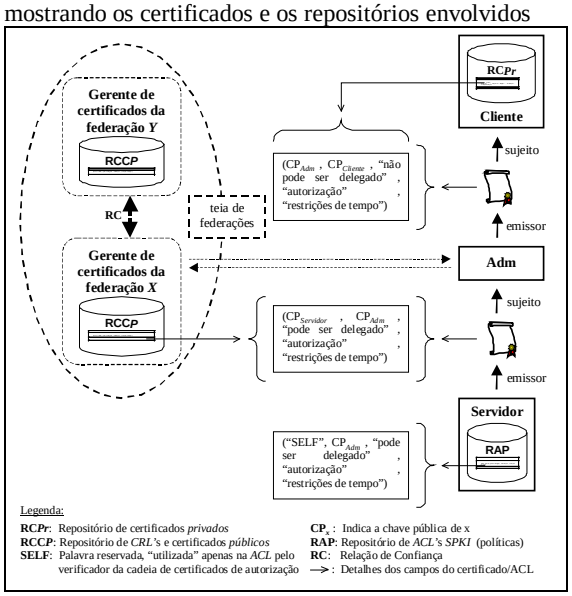
Supondo-se um cliente desejando se certificar que a chave privada assinando o primeiro certificado da cadeia de autorização, é mesmo do servidor em questão. Tal cliente deverá recuperar o certificado de nome do servidor, a partir da teia de federações, para verificar o nome do servidor. Em seguida, este cliente deve usar a chave pública do campo de emissor – do primeiro certificado da cadeia de autorização – para verificar a assinatura digital do certificado de nome do servidor. Se todas as verificações forem bem sucedidas a identidade do servidor estará verificada.

3.3 Autorização

No nosso modelo, preservou-se a estrutura disponível no *SDSI / SPKI* para a construção de *ACL*'s, descrevendo as políticas de autorização que regulam os acessos dos principais aos objetos locais ao servidor. Para o caso de aplicações distribuídas, certificados de autorização concedendo permissões de acesso por delegação são utilizados para alcançar o mesmo propósito. A concessão inicial de direitos de acesso é feita pelo guardião (servidor) do serviço e sua propagação se dá através de delegações sucessivas dos certificados de autorização, passando de um principal a outro e formando as cadeias de autorização (Figura 3.3).

As requisições de operações em um servidor devem ser assinadas pelo cliente e enviadas junto com a cadeia que o autoriza a realizar a operação (Figura 3.2). Quando verificada a autorização para efetuar a operação requisitada, o servidor emite um certificado atribuindo diretamente ao cliente, os direitos que lhe foram delegados através da cadeia de autorização (reduzida). Numa próxima requisição da mesma operação no servidor, este certificado é apresentado no lugar da cadeia de autorização.

Figura 3.3 – Cadeia de certificados de autorização, mostrando os certificados e os repositórios envolvidos



As políticas de autorização (*ACL's*) são locais ao servidor que oferece o serviço ou resultantes da concessão de privilégios feita por principais membros de uma cadeia, com direitos de delegação de privilégios – *trust* do servidor.

O processo de autorização é efetuado em nível de aplicação e o controle de acesso é executado pelo guardião. A confrontação da cadeia de autorização fornecida pelo cliente com a *ACL* que protege o recurso possibilita ao guardião tomar a decisão de permitir ou de negar um pedido de acesso (Tabela 3.2).

3.4 Auditoria

Para efeito de auditoria, são usados os registros (*log*s) de acesso das chaves públicas ao servidor. Quando for necessário será buscado o certificado de nome correspondente, na teia de federações, para identificar o principal detentor da chave pública que efetivou o acesso.

3.5 Formação de Novas Cadeias de Autorização

Na literatura científica há várias experiências com a busca de cadeias de certificados (seção 5), mas em todas quando a cadeia de certificados não é encontrada a busca termina com uma exceção (falha). Neste trabalho, através das federações, é criado um esquema que permite a um cliente localizar um privilégio de acesso dentro de uma teia de federações, o que dá um sentido efetivo de abrangência global aos certificados de autorização locais. Isto permite ao cliente negociar com o detentor de um privilégio a concessão do mesmo para formar novas cadeias de autorização.

Para dar suporte a formação de novas cadeias de autorização utilizou-se certificados de nome e de autorização, *threshold certificates* e grupos oriundos do *SDSI* / *SPKI* na efetivação da teia de federações. A seguir é mostrado como estes recursos são utilizados nas entidades do modelo para dar o apoio necessário ao processo de autorização e autenticação.

3.5.1 Gerente de certificados

O gerente de certificados define um grupo local congregando os filiados da federação. Para que um principal qualquer se filie a uma federação, é necessário apresentar um *threshold certificate* assinado por *k-de-n* membros (já filiados) a federação, acompanhado do certificado de nome do principal que está solicitando a admissão. O número (*k*) de membros necessários para endossar um pedido de filiação é definido por cada gerente de certificados da federação. O certificado de nome é guardado pelo gerente de certificados para auxiliar no processo de identificação de principais, como comentado nas seções 3.3 e 3.4. A todo novo membro aceito na federação é fornecido um certificado de grupo (certificado de nome expressando participação no grupo) para fins de comprovação da filiação.

O estabelecimento de relações de confiança entre federações (associação) também é interpretado como admissão de membro no grupo local das federações envolvidas. Só que neste caso, o novo associado (a outra federação e por conseguinte seus membros) é tratado como um grupo definido e administrado em outro espaço de nomes – de acordo com a definição de grupos prevista em *SDSI / SPKI*.

Como já mencionado anteriormente, além da mediação das relações de confiança entre membros e associados da federação, o gerente de certificados mantém a base de certificados consistente e fornece interfaces (padrão) para o armazenamento e recuperação de certificados. Especificamente para o caso da recuperação de certificados, o gerente implementa os algoritmos de busca das cadeias de nome e de autorização (uma variação do *DFS* para ambiente distribuído), que recorrem a teia de federações quando o certificado desejado não é encontrado localmente.

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 torna hierárquicas as relações de confiança e nem assume o papel de ponto crítico em relação a falhas e a vulnerabilidades ou ao desempenho do sistema.

3.5.2 Cliente

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, participa da formação de novas cadeias e da negociação para concessão dinâmica de privilégios de acesso (não discutida neste texto).

Como comentado na seção 3.1.2, o cliente possuirá certificados privados (Figura 3.3), que não são delegáveis ou mesmo que não julga conveniente publicar na federação. Assim, o cliente utilizará *um agente* (Figuras 3.1 e 3.2) para armazenar e recuperar estes certificados num repositório privado. O agente corresponde a um software executando atividades de automatização e de gestão dos certificados privados, em nome do cliente. A automatização compreende tarefas corriqueiras como verificação e efetivação de assinaturas, busca de cadeias de certificados, preenchimento de certificados de autorização (baseado em algumas regras predefinidas) e manutenção da consistência dos nomes locais. O agente permite também a adição de outras atividades que podem vir a ser automatizadas ou que precisam ser *online*. Assim, o agente está sempre ativo e o cliente se comunica com o mesmo através de um *binding* para uma interface de comandos.

Tabela 3.1 - Atividades envolvidas nas trocas entre cliente e servidor *SPKI*

Passo	Entidade(s) envolvida(s)	Descrição da atividade
I	C → S	O cliente solicita acesso a um recurso protegido (sem autenticação e nem autorização) para fazer uma operação (*OP*).
II	S → C	O guardião (servidor) devolve ao cliente um *challenge*, "prove a posse de privilégios para executar *OP* no objeto protegido pela *ACL*".
III	C	O cliente usa um algoritmo para descobrir se há uma cadeia de certificados de autorização (Figura 3.3) que lhe conceda privilégios para executar a operação *OP*.
IV	C → S	Se a cadeia existe, o cliente assina a requisição da operação *OP* e em seguida a envia junto com a cadeia de certificados de autorização ao guardião. Esta mensagem representa o *response* ao *challenge* proposto no passo I.
V	S	O guardião executa a seqüência de passos descrita na Tabela 3.2.
VI	S → C	Se todos os passos anteriores foram concluídos com sucesso, o servidor honra o acesso pleiteado pelo cliente.

As atividades envolvidas nas trocas entre cliente (C) e servidor (S), que estão sendo usadas nas requisições de acesso, estão baseadas no modelo de autorização adotado do *SPKI* (Clarke, 2001) e são mostradas brevemente na Tabela 3.1.

3.5.3 Servidor

O servidor de aplicação implementa o guardião e os objetos de serviço, os quais protege com *ACL*'s *SPKI*.

Um conjunto das atividades desempenhadas pelo servidor na efetivação do controle de acesso (passo V da Tabela 3.1) pode ser encontrado resumido na Tabela 3.2 (Clarke, 2001).

Tabela 3.2 - Seqüência de atividade executadas no controle de acesso do servidor de aplicação *SPKI*

Passo	Descrição da atividade
1	O guardião verifica o *timestamp* da requisição, para se certificar que a mensagem é recente.
2	O guardião cria o *tag* ("operação") correspondente a requisição do cliente para confrontá-lo com o que foi devolvido no *challenge* (passo II - Tabela 3.1.).
3	O guardião verifica se a chave privada que assinou a requisição é o par da chave pública contida no campo de sujeito da cadeia de certificados de autorização. Caso afirmativo a requisição é autêntica – isto representa a autenticação.
4	O guardião verifica a assinatura digital de cada certificado de autorização, utilizando a chave pública contida no campo de emissor de cada certificado verificado. Se todas as assinaturas forem autênticas e todos os períodos de validade dos certificados estiverem em vigência, então a seqüência é válida. Se este passo for bem sucedido significa que há um caminho de confiança ("uma seqüência de entidade confiáveis") por onde foram propagados os certificados de autorização até o cliente (sujeito). Este caminho de confiança tem uma função análoga a hierarquia de *CA*'s do *X.509*. Porém, neste casos, a construção da cadeia de autorização é totalmente arbitrária e depende apenas da confiança mútua entre os membros da mesma e não de uma entidade "central" de confiança.
5	Se o passo 4 for bem sucedido, o guardião verifica se a seqüência de certificados de autorização concede permissões suficientes ao cliente para a execução da operação desejada. Caso afirmativo, se tem a comprovação da autorização.

4 ASPECTOS DE IMPLEMENTAÇÃO DO MODELO PROPOSTO

A seguir são considerados aspectos de implementação do protótipo do modelo proposto.

4.1 Arquitetura do Protótipo

As ferramentas adotadas na composição da arquitetura do protótipo (Figura 4.1) são fortemente influenciadas pela freqüência de sua utilização na Internet – ambiente assumido como o contexto do trabalho. Porém, a infra-estrutura *SDSI / SPKI* e as políticas empregadas no modelo são totalmente independentes da tecnologia em uso e, em linhas gerais, representam as atividades descritas no modelo da seção 3.

No protótipo, basicamente, está se utilizando os protocolos *TCP / IP* (natos para o ambiente Internet) como infra-estrutura de comunicação e o *TLS* / *SSL* para dar segurança às mensagens em trânsito. O CORBA (*Common Object Request Broker Architecture*) e o CORBASec – serviços de segurança CORBA (OMG, 2001), estão sendo utilizados como *middleware*. O uso do *middleware* permite obter as funcionalidades de interoperabilidade e transparências – para as aplicações construídas a partir dos objetos distribuídos no ambiente

heterogêneo da Internet. Além disto, o CORBA e o CORBASec estão fundamentados numa especificação padronizada e na independência de ambiente de programação (o CORBA será discutido na seção 4.2). Os protocolos *http* e *ftp* foram mencionados apenas por compatibilidade com o ambiente Internet, mas poderiam ser quaisquer protocolos de aplicação. Os códigos Java, interpretados pelas *JVM*'s (*Java Virtual Machine* do cliente e do servidor), oferecem uma variedade de opções e facilidades ao desenvolvimento de sistemas amigáveis, para o usuário final, a partir de protocolos de aplicação na Internet.

Figura 4.1 – Arquitetura do protótipo de implementação do modelo

4.2 CORBA

A idéia fundamental do uso do CORBA, é usufruir do suporte a objetos distribuídos nos aspectos referentes a localização de objetos (resolução de nomes) e a segurança em invocações remotas. No caso, para que se estabeleça uma sessão entre cliente e servidor usando um canal seguro (com integridade e confidencialidade), é necessária a autenticação mútua dos principais (cliente e servidor). Porém, o *SPKI* utiliza chaves como principais e não nomes, assim são utilizados os certificados de nome *SPKI* traduzidos para o *SSL*, como prevê a *RFC 2693*, no estabelecimento das sessões seguras.

O modelo de autenticação e autorização proposto está sendo implementado em nível de aplicação *(Security Level 2* do *CORBASec*). Porém, para obter os benefícios almejados do modelo de segurança CORBA, um conjunto mínimo de objetos em nível de *ORB* foram mantidos. Para este caso, estão sendo utilizados os objetos de sessão: *PrincipalAuthenticator*, *SecurityManager*, *Credentials*, *SecurityCurrent* e *SecurityPolicy*.

As implementações de *ORB* que permitem a efetivação do ambiente CORBA descrito, nos aspectos desejados e em conformidade com as especificações da *OMG* são: o Adiron *ORBAsec SL2* (Adiron, 2000) e o IONA *ORBacus* (IONA, 2001). Em conjunto com os *ORB*'s está sendo utilizado como módulo *SSL* o *IAIK-iSaSiLk* (IAIK, 2000). Porém, nada impediria que outras implementações que obedeçam aos mesmos requisitos tivessem sido adotadas.

5 TRABALHOS CORRELATOS

A seguir são destacadas brevemente algumas experiências e seus aspectos notáveis.

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 a cadeia de interesse.

Em (Aura, 1998) 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 a 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, 1998) – mais especificamente, uma proposta para que o algoritmo *DFS forward* apresentasse melhores resultados.

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 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*.

5.1. Considerações sobre as propostas

Em linhas gerais pode-se observar nas experiências relatadas, que os trabalhos descritos

em (Nikander e Viljanen, 1998) e (Aura, 1998) foram concebidos para versões preliminares de *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 das *RFC*'s *2692* e *2693*. Já o trabalho de (Ajmani, 2000) é praticamente uma implementação resolvendo um problema de (Aura, 1998). Porém, em termos de arquitetura, nestas experiências boas idéias foram apresentadas.

Quanto ao trabalho de (Li, 2000), seu principal objetivo foi criar algoritmos de busca voltados para a programação em lógica. Portanto, nenhuma arquitetura foi proposta, mas sua interpretação de nomes locais como grupos distribuídos mostra-se como uma contribuição bastante importante.

Em (Clarke, 2001), o conteúdo da *RFC 2693* é refinado e um relato de implementação da versão atual de *SPKI* bastante rico em conteúdo é apresentado, mas não se tem uma proposta de arquitetura de abrangência mais ampla.

Em nosso trabalho se buscou principalmente propor uma arquitetura com uma concepção distribuída, o mais automática e descentralizada ("democrática") possível – através das teias de federações. Na integração com o ambiente, o mínimo de esforço é exigido do cliente, pois *plugin*s e/ou *applet*s tornam esta ação facilitada. Além disto, tão logo a implantação do modelo esteja concluída, a formação de novas cadeias de autorização (seção 3.5) poderá ser medida quanto a sua efetividade. Adicionalmente, um mecanismo de negociação para permitir a concessão dinâmica dos privilégios de acesso poderá ser proposto para automatizar ainda mais o processo de formação de novas cadeias.

6 CONCLUSÃO

O modelo de autorização e autenticação proposto tenta minimizar os impactos da rigidez do modelo baseado num universo global de nomeação. Através da adoção das redes de confiança e da construção das teias de federações, os certificados de nome e de autorização locais ganham um sentido global.

Deve-se considerar que pelo estágio atual de implantação do modelo, não se pode precisar o tempo de resposta e nem o esforço computacional necessário para processar as busca distribuídas das cadeias de certificados, tanto de nomes quanto de autorização, quando a teia de federações ganhar abrangência global.

Por mais que os aspectos de desempenho do sistema ainda não tenham sido avaliados, acreditamos que este modelo se mostra mais promissor do que as experiências relatadas até o momento, devido principalmente a adoção de padrões e técnicas que visam o ambiente distribuído de larga escala.

REFERÊNCIAS BIBLIOGRÁFICAS

ADIRON, LLC. *ORBAsec SL2: User Guide*. Version 2.1.4, July 2000

AJMANI, Sameer. *A trusted Execution Platform for Multiparty Computation*. Master thesis, Department of Eletrical Engeneering and Computer Science, MIT, 2000.

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

BLAZE M., e outros. *The KeyNote Trust Management System, Version 2, RFC2704*. IETF, 1999.

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

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

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

ELLISON, C. e outros. *SPKI Certificate Theory*. *RFC2693*, IETF, 1999.

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

HASTINGS, Nelson E. e POLK, W. Timothy. *Bridge Certification Authorities: Connecting B2B Public Key Infrastructures. NIST*, 2000.

HORST F. Wedde, Mario LISCHKA, *Modular Authorization*. In: *SACMAT'01*,2001.

IAIK (Institute of Applied Information and Communication). *ORBAsec SL2: User Guide*. Version 2.1.4, July 2000.

IONA Technologies, Inc., *ORBacus: User Guide*. Version 4.1.0, 2001.

LAMPSON, B. and RIVEST, R. L., *A simple Distributed Security Infrastructure*. http://theory.lcs.mit.edu/~cis/sdsi.html, 1996.

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

NIKANDER, Pekka e VILJANEN, Lea. *Storing and Retrieving Internet Certificates*. In: 3th Nordic Workshop on Secure IT Systems, 1998.

OMG (*Object Management Group*). *Security Service, v1.7*. http://www.omg.org/cgi-bin/doc?formal/2001-03-08, 2001.