Transferência de autenticação e autorização através de Serviços *Web*

Emerson Ribeiro de Mello*, Joni da Silva Fraga , Edson Tavares de Camargo

Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
88040-900 Florianópolis, SC

{emerson,fraga,camargo}@das.ufsc.br

Abstract. *This work presents a model that guarantees authentication and authorization transfer among differents security and administrative domains. The model uses security specifications for Web Services architecture and is based on the federation web concept, which allows scalable and flexible rights management solutions. In this work is presented an example, that illustrates the integration of different security technologies, in the case, the SPKI/SDSI and X.509.*

Resumo. *Neste trabalho é apresentado um modelo que visa permitir a transferência de autenticação e autorização entre diferentes domínios administrativos e de segurança. O modelo faz uso das especificações de segurança propostas para a arquitetura dos Serviços Web e está fundamentado no conceito das teias de federações que permite soluções de gerenciamento de direitos escaláveis e flexíveis. No trabalho é apresentado um exemplo, que ilustra a integração de diferentes tecnologias de segurança, no caso, o SPKI/SDSI e o X.509.*

1. Introdução

Transparente para plataformas e com um modelo fracamente acoplado são características que tornam Serviços *Web* (*Web Services*) [W3C 2004] ideais como tecnologia integradora, permitindo que aplicações distribuídas sejam construídas e mantidas de maneira flexível e rápida em ambientes de larga escala, como a Internet. A facilidade para transposição de filtro de pacotes (*firewalls*), antes proibitiva para aplicações distribuídas, por exemplo, desenvolvidas em CORBA [OMG 2002], também é considerada como um fator de alto risco, quando a segurança é um requisito considerado.

Outro aspecto em relação ao modelo dos Serviços *Web* que coloca dificuldades em termos de segurança é que as aplicações servidoras não se limitam a interações diretas com seus clientes. Serviços *Web* podem servir de um simples retransmissor para as requisições de clientes que seriam roteadas a nível de aplicação para outros serviços, os verdadeiros implementadores das operações desejadas. Os acessos intermediários são feitos em nome do principal[1] que originou o pedido inicial.

Com o objetivo de tornar o uso dos Serviços *Web* seguro e assim garantir a sua ampla adoção, muitas propostas de segurança estão sendo submetidas aos órgãos como W3C

*Bolsista CNPq.

[1]Usuário, processo ou máquina autorizados pelas políticas do sistema.

(*World Wide Web Consortium*) e OASIS (*Organization for the Advancement of Structured Information Standards*) e em conjunto com as especificações de extensões de segurança do XML, as propostas visam cobrir diversas áreas de segurança. Tecnologias de camadas subjacentes também podem ser utilizadas em conjunto para prover uma maior segurança, como por exemplo o SSL (*Secure Sockets Layer*) [Freier et al. 1996].

O gerenciamento das aplicações distribuídas construídas segundo o modelo Serviços *Web* também é um grande desafio, uma vez que os limites administrativos são atravessados, estas estarão sob diversos modelos administrativos, integrando diversas tecnologias de implantação e ainda envolvendo diversos mecanismos e modelos de segurança. Cada domínio de segurança atravessado por uma aplicação distribuída pode prover seu próprio conjunto de credenciais de segurança, tomando como base suas tecnologias de segurança subjacentes e políticas.

Este artigo apresenta um modelo de confiança para aplicações distribuídas orientadas a serviços. O modelo assume premissas de autenticação e autorização atravessando diversos domínios administrativos e de segurança. Este modelo deve servir de mediador entre esquemas de confiança de diferentes domínios de segurança, envolvendo clientes e serviços dispostos segundo a aplicação distribuída. Portanto, os mecanismos para a localização de direitos em ambientes heterogêneos, deve tratar com diferentes tecnologias de segurança. Estas tecnologias expressam normalmente direitos e controles de forma diversa e não interoperável. O papel deste modelo é permitir, por exemplo, a interação entre um cliente em um domínio que faz uso do X.509 [ITU-T 1993] com um provedor que tenha seus controles implementados baseados em certificados SPKI/SDSI (*Simple Public Key Infrastructure / Simple Distributed Security Infrastructure*) [Ellison et al. 1999].

Este trabalho está organizado da seguinte forma. As propostas de segurança para Serviços *Web* são apresentadas na seção 2.. A seção 3. apresenta a idéia do agrupamento de principais através do conceito de federações e o modelo de confiança para Serviços *Web*. Detalhes sobre a implementação estão contidos na seção 4.. Na seção 5. é feita uma discussão sobre os trabalhos relacionados. E por fim, na seção 6. são dadas as conclusões.

2. Padrões de segurança para Serviços *Web*

Nesta seção serão apresentadas especificações sobre segurança destinadas à arquitetura dos Serviços *Web*. Serão apresentados padrões de segurança em XML que também são utilizados no modelo proposto e nas especificações para Serviços *Web*.

2.1. WS-Security

Padronizada pela OASIS, a *WS-Security* [OASIS 2004b] descreve melhorias e extensões nas mensagens SOAP para agregar as propriedades de confidencialidade e integridade. É possível utilizar uma grande variedade de mecanismos de segurança e tecnologias para cifragem, como por exemplo infra-estruturas de chave pública (ICP), Kerberos [Kohl and Neuman 1993] ou SSL [Freier et al. 1996]. Essa especificação foi projetada para ser expansível, permitindo o uso de múltiplas credenciais, possibilitando ao cliente utilizar diferentes formatos de credenciais para a autenticação e autorização.

2.2. WS-Policy

A especificação WS-Policy [WS-Policy 2004] provê uma gramática flexível que possibilita expressar políticas em sistemas baseados em XML. A especificação inclui um

conjunto de asserções gerais, relacionadas às mensagens SOAP; estas asserções são definidas pela especificação WS-PolicyAssertion [WS-PolicyAssertions 2003]. Asserções relacionadas à segurança e que suportam a especificação WS-Security, são tratadas na especificação WS-SecurityPolicy [WS-SecurityPolicy 2005]. A definição de qual algoritmo usar para cifrar e assinar as requisições a um determinado Serviço *Web* seria uma das habilidades descritas pela WS-SecurityPolicy.

A WS-Policy não descreve como essas políticas são divulgadas ou como anexá-las a um Serviço *Web*, assunto coberto pela especificação WS-PolicyAttachment [WS-PolicyAttachment 2004], que define como anexar as políticas com elementos XML, WSDL [Christensen et al. 2001] e UDDI [OASIS 2004a].

2.3. WS-Trust

A WS-Trust [WS-Trust 2005] é uma especificação que, como a WS-Security, tem o sentido de prover um ambiente seguro para os Serviços *Web*. Trata-se de um esforço inicial que visa principalmente a troca de credenciais de segurança, para que possibilite a comunicação através de diferentes domínios de segurança. A especificação descreve as relações de confiança que podem ser estabelecidas:

- Direta – o domínio A confia nos atributos oriundos do domínio B;
- Mediada – os domínios A e B possuem relações de confiança com um domínio C, mas não possuem relações entre si.

A segurança no WS-Trust é baseada na premissa que qualquer pedido recebido por um Serviço *Web* contenha um conjunto de atributos de segurança (ex: nome, chave, direitos, etc.). Os Serviços de Atributos de Segurança (*Security Token Services* - STS) são introduzidos na especificação WS-Trust como as autoridades responsáveis por emitir os atributos de segurança. Estas entidades requerem seus próprios atributos de autenticação e autorização que, com o uso dos mesmos, formam a base de confiança do modelo.

2.4. Tecnologias XML

A arquitetura dos Serviços Web está diretamente ligada ao XML e as extensões de segurança do mesmo, como o caso da XML-Signature [Bartel et al. 2002] e da XML-Encryption [Imamura et al. 2002], que expressam assinaturas e cifragem em XML.

A especificação SAML (*Secure Assertion Markup Language*) [OASIS 2002] é uma infra-estrutura projetada para expressar informações sobre autenticação e autorização. O SAML não provê a autenticação em si, mas sim meios para expressar informações de autenticação que ocorreram no sistema. Baseado no princípio da "confiança portável", o seu uso combinado com a especificação WS-Trust (veja seção 2.3.) provê a possibilidade de atravessar domínios administrativos com uma única autenticação (*Single Sign-On*), por exemplo.

O XKMS (*XML Key Management Specification*) [Ford and Hallam-Baker 2001] é um padrão aberto desenvolvido para retirar a complexidade de trabalhar com infra-estruturas de chave pública dos clientes. O XKMS provê dois subprotocolos: *XML Key Information Service Specification* – X-KISS, para a localização e validação de chaves, certificados, etc.; e o *XML Key Registration Service Specification* – X-KRSS, responsável para o registro, revogação, renovação e recuperação de chaves.

3. Modelo de confiança para ambientes orientados a serviços

Nesta seção, introduzimos o modelo de confiança definido para aplicações distribuídas desenvolvidas através de Serviços *Web*. O modelo, por ter como base uma tecnologia integradora, deve assumir premissas de autenticação e autorização atravessando diversos domínios administrativos, por onde se estende a aplicação distribuída. Diante disto, uma premissa fundamental é que este modelo deva servir de mediador entre modelos de confiança de diferentes domínios de segurança.

As especificações WS-Trust e WS-Federation [WS-Federation 2003] fornecem conceitos, serviços e protocolos que formam a base deste modelo de confiança desenvolvido para ultrapassar limites administrativos e domínios de segurança. Porém, estas especificações são omissas no que se refere a dinâmica da formação de relações de confiança em ambientes normalmente heterogêneos e complexos. O nosso modelo está fundamentado no conceito de *federações* [Santin et al. 2003, WS-Federation 2003, Liberty 2003] como forma de agrupar usuários e no sentido de facilitar a escalabilidade de nossas soluções na gestão das relações de confiança. Cada federação em nosso modelo caracteriza um domínio de segurança.

3.1. Federações

Em ambientes compostos por diferentes tipos de indivíduos, cada qual com diferentes interesses, o gerenciamento dos mesmos é uma tarefa árdua, quando considerado um ambiente de larga escala. A forma clássica para facilitar a administração é sempre agrupá-los de acordo com suas habilidades e interesses. Em ambientes como a Internet o problema é a forma de dispor estes grupos e a relação entre os mesmos. Para aumentar a complexidade, indivíduos podem ainda estar presentes em mais de um grupo. Por exemplo, clientes de uma livraria, dispostos na forma de um grupo, poderiam pagar suas contas através de cheque ou cartão eletrônico, estes oriundos de um outro grupo, no caso o grupo de clientes de um banco. Assim, para obter escala e atingir todos os indivíduos e serviços é preciso que grupos se comuniquem e tenham relações de confiança entre si.

O modelo de confiança proposto em [Santin et al. 2003] é baseado em Federações SPKI, e apresenta como finalidade a resolução de cadeias de certificados de autorização SPKI/SDSI e o estabelecimento dinâmico de novas cadeias de certificados. Cada federação funciona de forma semelhante a um repositório passivo de certificados. Um principal, ao ingressar na federação, fornece todos os certificados de autorização que achar pertinente delegar, para que outros principais possam usufruir das permissões que este possui. A escalabilidade no ambiente é conseguida através das associações entre as federações (teias de federações). Tais associações permitem que principais consigam efetuar buscas por estas teias de federações, sem a necessidade da filiação em inúmeras federações. É um sistema igualitário de confiança que não impõe hierarquias de chaves para ganhar em escala como as formadas pela ICP do X.509.

O agrupamento de entidades (serviços e clientes), através de federações, apresentado nas especificações [WS-Federation 2003] e [Liberty 2003] visa diminuir a complexidade da gerência de identidades de clientes e provedores de serviço, porém sem que haja a necessidade de um repositório central para o armazenamento dessas identidades.

3.2. Aspectos estruturais de um Domínio *Web*

O modelo proposto está fundamentado no conceito de *federação* e nas associações de federações permitindo a construção das "*Teias de Domínios Web*". No modelo de confiança proposto, cada Domínio *Web* é composto por um gerente que agrupa os seus diversos filiados através de seus atributos de segurança (credenciais, certificados, etc). As características destes gerentes vai depender da tecnologia de segurança subjacente. Por exemplo, se este gerente estiver encapsulando a infra-estrutura SPKI/SDSI, o mesmo se torna um simples repositório de certificados autorização e de nomes desta ICP. Se este, ao contrário, corresponder a um servidor Kerberos [Kohl and Neuman 1993], então os serviços de autenticação e de fornecimento de *tickets* para conexões seguras (*Ticket Granting Service*) deste servidor estarão disponíveis através deste gerente.

Ou seja, qualquer tecnologia de segurança é representada a partir do gerente de um Domínio *Web*. Em qualquer destas tecnologias de segurança, o gerente mantém o controle dos membros, gerenciando o ingresso e egresso, bem como as consultas realizadas por estes. As interfaces deste gerente com o mundo dos Serviços *Web* passam pelos serviços STS e XKMS.

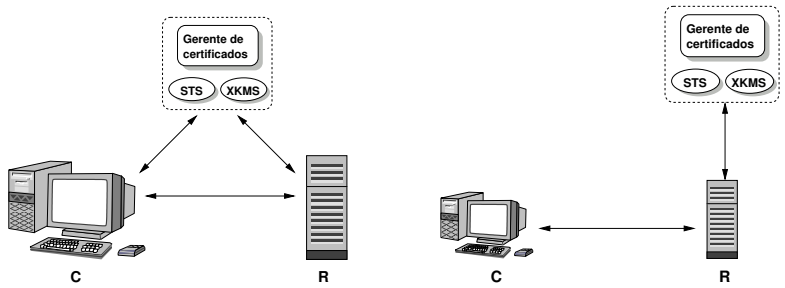
O STS é o responsável por emitir ou garantir os atributos que são válidos para todos os participantes da relação de confiança, o estabelecimento dessa relação de confiança deve ser atingido através da nossa proposta. O serviço STS tem papel fundamental no nosso modelo principalmente na intermediação de relações de confiança envolvendo dois diferentes domínios de segurança. Um principal pode requisitar através do STS atributos de segurança (que podem ser, conforme a tecnologia de segurança, *tickets*, cadeias de certificados, etc.) necessários para que possa efetuar acesso a um serviço contido em um outro domínio de segurança.

O serviço XKMS permite a localização e validação de chaves e atua como um agente que busca tirar a complexidade do cliente, no trato com uma infra-estrutura de chave pública. No modelo, o XKMS pode ser usado na busca e validação de cadeias de certificados SPKI, por exemplo. A especificação não se preocupa com qual infra-estrutura de chave pública será feita disponível pela interface XKMS, não descrevendo como chaves, certificados, etc. poderão ser recuperados/validados. Para o caso de usar o X.509 [ITU-T 1993] a infra-estrutura poderia ser simplesmente um repositório de chaves e certificados acessado pelo XKMS, com uma relação hierárquica de Autoridades Certificadoras válidas, disponíveis através de comunicações entre STS. No modelo SPKI/SDSI não são consideradas Autoridades Certificadoras, cada principal está apto a emitir e assinar certificados e as relações de confiança são estabelecidas de acordo com a política de negócio de cada principal. Por ser um modelo distribuído, sua maior dificuldade está na localização dos detentores de direitos. Para suprir essa dificuldade, lançamos uma heurística (seção 3.5.) que descreve a navegação na teia de domínios na tentativa de localizar o detentor do direito desejado para a possível delegação do mesmo se for o caso.

3.3. Relações de confiança em um Domínio *Web*

Como descrito anteriormente (seção 3.2.) os Domínios *Web* são compostos por um gerente e diversos filiados (principais), existindo assim uma simples relação de confiança dos afiliados com o gerente do Domínio. Credenciais de segurança emitidas pelo STS do gerente são consideradas confiáveis por todos os afiliados.

A figura 1 ilustra algumas formas de delegação de direitos envolvendo somente um Domínio *Web*. No caso ilustrado pela figura 1(a), um cliente "C" deseja invocar um Serviço *Web* provido por "R". Ao receber o pedido, "R" analisa se o mesmo está acompanhado das asserções SAML (seção 2.4.) necessárias. Caso não, o cliente "C" é informado da política de acesso aplicada ao serviço – expressa em WS-Policy – indicando quais são os atributos necessários para realizar a invocação (o acesso). Com o desafio em mãos o cliente aciona (mensagem `wstrust:RequestSecToken`) o serviço STS do gerente do seu domínio o qual, através de uma política de negócios, lhe concede as asserções necessárias (mensagem `wstrust:RequestSecTokenResp`). Munido das asserções, o cliente "C" responde o desafio a "R" e, este depois de verificar que as asserções apresentadas são válidas e suficientes, garante o acesso ao serviço[2].



(a) Cliente ativo nas invocações

(b) Cliente passivo nas invocações

Figura 1. Relações de confiança inter-domínio

No caso apresentado na figura 1(a), "R" possui uma relação de confiança com o gerente do seu domínio e assim, asserções emitidas pelo gerente são ditas confiáveis. Esse tipo de relação de confiança pode ser classificado como "Raiz de confiança fixa". Trata-se de um caso onde o cliente é ativo, ou seja, está apto a tratar desafios fornecidos pelo serviço.

Na figura 1(b), "R" ao receber uma requisição e verificar que a mesma não contém as asserções necessárias, ao invés de enviar um desafio para "C", aciona o gerente do domínio para que ele gere as asserções necessárias para "C" e assim garantindo o acesso ao serviço. Neste caso, a aplicação cliente assume o papel passivo, não estando apta a tratar desafios, necessitando que os mecanismos para autenticação e autorização sejam transparentes a mesma.

Ambos os casos (Cliente passivo ou ativo) poderiam ser aplicados em um domínio fechado, o qual é composto por diversos clientes e diversos provedores de serviços. O gerente do domínio estaria assumindo um papel de entidade autenticadora, garantido a cada cliente (principal), uma identidade autenticada associada com direitos de acesso. Qualquer cliente de um domínio, para as comunicações com as interfaces STS e XKMS do seu gerente, terá que importar *stubs* correspondentes permitindo a execução dos respectivos protocolos.

[2]Para garantir proteção contra ataque de mensagens antigas, um número aleatório que só é utilizado uma única vez (*nonce*) acompanha o protocolo.

3.4. Relações de confiança envolvendo diferentes domínios

A proposta apresentada neste trabalho visa a interoperabilidade entre diferentes Domínios *Web*, os quais poderão usar diferentes tecnologias de segurança. O caso apresentado a seguir é composto por três Domínios *Web* (veja figura 2). O domínio 1 utiliza como tecnologia de segurança o SPKI/SDSI. O domínio 3 está fundamentado sobre o X.509. O gerente do domínio 2 possui habilidade em trabalhar tanto com SPKI/SDSI quanto com X.509 e, no exemplo apresentado, intermediará a confiança entre os domínios 1 e 3, visto que não existe uma relação de confiança entre 1 e 3.

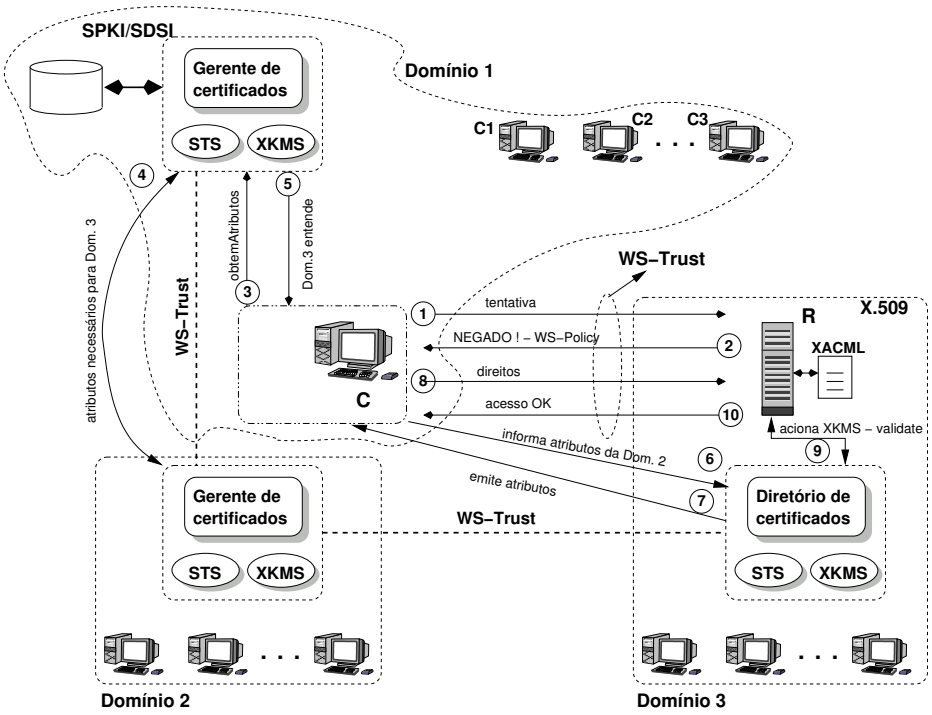


Figura 2. Iteração entre participantes no modelo proposto

O cliente “C”, no domínio 1, está pleiteando o acesso ao serviço provido por “R” (passo 1), este presente no domínio 3. O desafio lançado por “R” poderia ser do tipo “aceito asserções emitidas pela chave do principal *X*” (passo 2). O cliente “C” deve procurar *X* acionando o serviço XKMS do seu gerente para que o mesmo localize o respectivo principal.Com esta localização, “C” poderá através de uma comunicação direta com o detentor dos direitos (principal *X*), obter a delegação dos mesmos. Este é o modelo típico da ICP SPKI/SDSI. Através de um repositório central de direitos dentro do domínio, o gerente auxilia os membros na localização de direitos.

Há casos onde o principal detentor dos direitos não pertence ao domínio SPKI/SDSI e assim, não está apto a delegar certificados de autorização. Supondo que “R” só aceite atributos de segurança emitidos pelo gerente de seu próprio domínio. No caso do exemplo da figura 2, o desafio enviado por “R” pode conter a localização do seu gerente (gerente do domínio 3) (passo 2). Porém, o cliente “C” não poderá acioná-lo, visto que o cliente, em nosso exemplo, é um principal SPKI/SDSI e desconhece o funcionamento do X.509, ICP que fundamenta o gerente de “R”. Dessa forma, o cliente “C” aciona o gerente do seu domínio para tentar achar um “caminho de confiança” que lhe garanta autorização ao serviço “R”(passo 3).

Os gerentes de “C” (Domínio 1) e de “R” (Domínio 3) não possuem uma relação de confiança entre si. O gerente de “C” deve realizar uma busca (heurística, seção 3.5.) por todos os domínios com quem possui uma relação de confiança, com o intuito de encontrar um caminho de confiança que o leve até o gerente de “R”. Em nossa proposta essa busca realizada pelos gerentes possui um funcionamento semelhante ao protocolo Gnutella [Gnutella 2001], o que facilita a navegação pela teia de associados, conseguindo assim obter escala sem haja necessidade de um gerente conhecer todos os outros possíveis parceiros da teia.

No exemplo apresentado na figura 2 o caminho entre o Domínio 1 e 3 é intermediado[3] pelo Domínio 2. O gerente de “C” solicita ao STS[4] do domínio 2 os atributos necessários para que se possa estabelecer uma comunicação com o gerente de “R” (passo 4). Por fim, o gerente do domínio 1, através do seu serviço STS, fornece ao cliente “C” todas as credenciais necessárias (emitidas por ele + emitidas pelos gerentes intermediários) para que o mesmo possa comunicar-se com o serviço STS do gerente de “R” (passo 5). Assumindo uma negociação simples, o gerente de “R” analisa as credenciais informadas, verifica se as mesmas são válidas e então fornece (no passo 6) as credenciais solicitadas pelo cliente. Em posse dos direitos, o cliente envia a resposta do desafio para “R” (passo 8), onde o mesmo efetua o confronto com as políticas aplicadas a ele e, a validação das assinaturas usando o serviço XKMS de seu domínio (passo 9), garantindo assim acesso (passo 10). Maiores detalhes sobre o controle de acesso, aplicado no recurso “R”, serão mostrados na seção 3.6..

No caso apresentado acima o gerente do domínio 1 tem papel fundamental e ativo. É ele quem procura pelos gerentes associados, solicita as credenciais e fornece ao cliente toda a base necessária para que o mesmo consiga comunicar-se diretamente com o detentor dos direitos, no caso o gerente de “R”. Esse tipo de comportamento permite deixar menos complexa a implementação dos membros de cada domínio *web*, tornando essa busca transparente para os mesmos. Porém, uma outra lógica plausível e interessante seria o caso onde o gerente localizaria o caminho de associações e retornaria essa informação ao cliente, sendo este então o único responsável por negociar com cada gerente intermediário até conseguir obter os direitos para se comunicar com o serviço desejado.

A confiabilidade do canal em relação a integridade e confidencialidade, é atingida através da WS-Security bem como de asserções SAML, não havendo a necessidade (mas não proibindo) do uso de mecanismos nas camadas inferiores, como por exemplo o uso do SSL.

3.5. Heurística da busca por direitos

No caso do exemplo ilustrado pela figura 2, o gerente do domínio 1 deseja localizar, entre os domínios com quem possui relação de confiança, um caminho de confiança que o leve até ao gerente do domínio 3. É apresentado aqui uma heurística de buscas que em nossos trabalhos foi concebida para o modelo de interação par a par (*peer-to-peer*). O protocolo

[3]Sabendo que poderão existir inúmeros domínios intermediários, sendo que quanto maior for esse número de intermediários, menos provável será a obtenção dos direitos desejados, devido ao fato da complexidade envolvida.

[4]A política de negócios envolvida para a emissão desses atributos está relacionada diretamente com a aplicação, podendo ser simples, onde cada pedido gera uma resposta sem custos, ou complexa, o que envolveria pagamento de taxas.

que segue este modelo possui duas mensagens: *query*, que é usada para efetuar a busca por recursos; e a *queryHit*, informando que o recurso procurado foi encontrado. O algoritmo abaixo descreve como a busca por recursos é realizada, no caso a mensagem "*query*".

Algoritmo 1 $query(origem, recurso, P, ttl)$

Require: $T = \{$ Tabela com todos os nós com quem possui relações de confiança $\}$
Require: $D = \{$ Diretório local sobre informações dos recursos providos $\}$
1: **if** $(recurso \subset D)$ **then**
2: $queryHit(origem, recurso, P, p)$
3: **else**
4: **if** $(ttl > 0)$ **then**
5: $N \longleftarrow T$
6: **while** $N \neq \varnothing$ **do**
7: $X \longleftarrow firstElement(N)$
8: $P \longleftarrow origem \cap P$
9: $query(noAtual, recurso, P, ttl - 1)$
10: $N \longleftarrow N \setminus X$ {Remove o conjunto X do conjunto N}
11: **end while**
12: **end if**
13: **end if**

A mensagem *query* é composta por quatro parâmetros: *origem* – de onde partiu a invocação; *recurso* – informando qual o recurso a ser procurado; *P* – um conjunto contendo a seqüência reversa de todos os nós por onde passou a requisição; e *ttl* - a qual indica um tempo de vida para a procura, proibindo que a mesma se estenda indefinidamente, limitando assim a sua propagação.

Um nó (*p*), em nosso caso um gerente de domínio na teia, ao receber a mensagem "*query*" verifica em seu repositório local – conjunto *D* do algoritmo 1 – se o mesmo possui o "*recurso*" procurado (um caminho de confiança) e em caso afirmativo envia uma mensagem "*queryHit*" para o nó que originou a mensagem "*query*". Caso contrário, é enviada uma mensagem "*query*" para todos os nós (domínios) – conjunto *T* – com quem possui relação de confiança (linha 6 – 11). Para cada novo nível que a mensagem "*query*" desce, o valor da variável "*ttl*" é decrementado, evitando que a mensagem se propague indefinidamente.

3.6. Controle de acesso

Esta seção detalha o passo 9 da figura 2, ilustrando o funcionamento do processo de controle de acesso aplicado no recurso "R". As políticas de controle de acesso são definidas em XACML [OASIS 2005] e a decisão e a aplicação das mesmas são feitas através do PDP (*Policy Decision Point*) e PEP (*Policy Enforcement Point*) [Yavatkar et al. 2000], respectivamente. A figura 3 ilustra os passos envolvidos no controle de acesso.

Sempre que um cliente realizar um pedido sem fornecer asserções de autenticação emitidas por uma autoridade confiável, este pedido é interceptado pelo PEP, o qual repassa à Autoridade de Autenticação (passo 1) que (por sua vez) faz uma consulta ao serviço X-KISS – provido pelo interface XKMS do gerente do domínio – a fim de validar a assinatura (passo 2). No passo 3 é gerada uma asserção SAML de autenticação (passo 3).

Com posse da asserção SAML de autenticação, o PEP fornece a mesma à Autoridade de Atributos (passo 4) com o intuito de obter uma asserção SAML de atributos. Na asserção são inseridos todos os atributos que estão relacionados a asserção SAML de autenticação fornecida pelo PEP. Por fim, é emitida uma asserção SAML de atributos (passo 5).

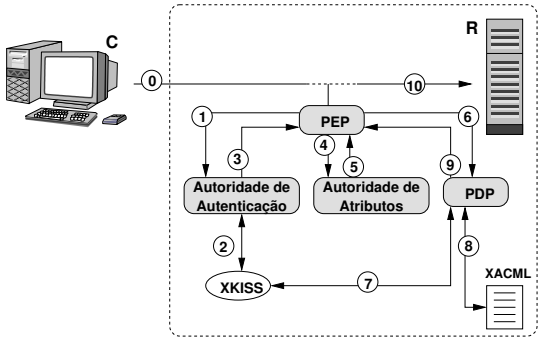


Figura 3. Controle de acesso no recurso

No passo 6 o PEP encaminha as asserções SAML de autenticação e de atributos ao PDP, para que o mesmo determine se o acesso deverá ser concedido ou não. O PDP confronta os atributos fornecidos com as políticas do sistema (passo 8) e retorna ao PEP uma asserção SAML de autorização (passo 9). O PEP em posse da asserção SAML de autorização, determina as informações que o cliente poderá obter do recurso protegido (passo 10).

4. Implementação

Para a implementação foi utilizada a linguagem Java e como servidor de aplicação foi adotado o TomCat versão 5. Como implementação SOAP foi utilizada o Apache Axis e para trabalhar com cifragem e assinatura de documentos XML foi utilizada a API Apache XML Security. E para o suporte ao SAML foi utilizada a API OpenSAML.

O protótipo implementado consiste de Serviços *Web* onde foi definido um domínio com um gerente SPKI o qual permite que relações de confiança sejam estabelecidas dinâmicamente, dando meios para a localização e negociação dos direitos com os respectivos membros detentores. Foram implementados também os serviços STS e XKMS do gerente do domínio, atendendo requisitos de interoperabilidade visto que os serviços STS do nosso protótipo deverão se comunicar com diferentes gerentes que encapsulam diferentes tecnologias de segurança (no protótipo estamos prevendo inicialmente as ICPs X.509 e SPKI/SDSI), para possibilitar a intermediação entre tecnologias. O desenvolvimento do serviço XKMS se fez necessário visto que as atuais implementações, como a TSIK (*Trust Service Integration Kit*) da VeriSign [VeriSign 2002], não possuem suporte para certificados SPKI/SDSI.

Como aplicação exemplo foi utilizada um trabalho anterior nosso [Ancajima et al. 2004], chamado “Serviço para busca de artigos”. Essa aplicação trata a necessidade atual da existência de um repositório central sobre trabalhos acadêmicos. Geralmente, cada organização possui seus próprios mecanismos para o armazenamento e localização de artigos, o que obriga aos usuários destes sistemas a aprenderem como utilizar cada sistema. A aplicação de exemplo atua como um repositório central que possui uma interface que permite realizar buscas informando nomes dos autores, título do trabalho entre outras. Assim, os usuários do sistema só precisarão se autenticar uma única vez, realizar uma única busca e obter resultados oriundos de diferentes instituições.

A aplicação consiste de uma interface única e simples a qual reúne informações

provenientes de diferentes provedores de serviço. Para o usuário do sistema, essa distribuição é transparente e os mecanismos de controle de acesso entre os diferentes sistemas cooperam através de relações de confiança entre os domínios, permitindo assim que atributos de segurança do cliente emitidos em um domínio possam ser reconhecidos em outros domínios. O ambiente envolvido pelo protótipo consiste de três domínios, onde a relação de confiança entre o domínio do cliente com o domínio do provedor do serviço é intermediada através de um terceiro domínio (conforme ilustrado pela figura 2).

5. Trabalhos relacionados

No trabalho de [Wlech et al. 2003] é descrito como permitir a criação dinâmica de serviços bem como de domínios de confiança, tendo sua aplicação voltada para o kit de ferramentas da Globus (uma plataforma para computação em grade) [Foster and Kesselman 1999]. A especificação da infra-estrutura de segurança para a computação em grade, assume a integração com os Serviços *Web* e assim usufrui dos padrões de segurança, como o SAML e WS-Security.

São mostrados alguns desafios de segurança presentes nas grades computacionais, sendo a dinâmica do ambiente o principal desafio, visto que serviços (recursos) podem ser ativados/desativados dinamicamente durante o ciclo de vida de uma sessão de alocação de recursos. Esse tipo de ambiente reúne diversos domínios administrativos e de segurança, em conseqüência, diferentes tecnologias de segurança. Na proposta, a segurança é provida na forma de serviços, sendo o *Serviço para Conversão de Credenciais* responsável por possibilitar que diferentes domínios comuniquem-se.

Os serviços de segurança descritos por Wlech [Wlech et al. 2003] possuem um funcionamento semelhante com os serviços empregados em nossa proposta, porém Wlech não descreve como as relações de confiança são estabelecidas e nem como localizar os possíveis detentores dos direitos, caso seja necessário. Em nosso trabalho essas questão são abordadas e o uso na arquitetura as grades computacionais poderia ser adotada sem grandes modificações.

O trabalho [Winslett et al. 2002] propõe uma arquitetura para o estabelecimento de relações de confiança entre partes estranhas através da revelação gradual das credenciais. Um caso típico faria uso de uma terceira parte confiável para que a negociação entre as partes possa ocorrer. Tal solução torna-se um gargalo em ambientes de larga escala. Segundo Winslett [Winslett et al. 2002], o uso de provas de conhecimento zero é difícil de ser implementado de maneira eficiente. Sendo assim, optaram pela maneira "o que precisa ser conhecido", onde as partes envolvidas disponibilizariam suas políticas somente quando for necessário.

A proposta consegue proteger as credenciais das partes envolvidas, porém sem que proíba a comunicação. Mas, caso a negociação envolva diversas partes, para cada parte será preciso estabelecer uma autenticação, causando um problema em um ambiente de larga escala. Assim sendo, uma solução que poderia ser adotada seria o uso de uma única autenticação, facilidade provida, por exemplo, pelo SAML.

A revelação parcial proposta por Winslett em tese se assemelha com a arquitetura proposta por nós. Cada principal é detentor de inúmeros direitos, esses expressos através de certificados SPKI/SDSI, ou através de outras formas. Os direitos só são revelados

de acordo com a necessidade que cada serviço. Esses direitos são expressos através de asserções SAML, o que possibilita ainda que os mesmos transponham domínios.

Em [Skogsrud et al. 2003] é apresentado um modelo dirigido a negociação de confiança para Serviços *Web*. O modelo define serviços que permitem interação entre autoridades de atributos e infra-estruturas de chave pública. Máquinas de estado são utilizadas para o gerenciamento do ciclo de vida das políticas, associadas aos recursos.

Como no trabalho de Winslett [Winslett et al. 2002], a proposta de Skogsrud [Skogsrud et al. 2003] visa obter o estabelecimento da confiança de forma gradual, fazendo uso das máquinas de estado, onde diferentes estados estariam associados a diferentes direitos. A proposta permite mudança das políticas correntes sem causar a interrupção das negociações em andamento.

A abordagem de Skogsrud de mapear em máquina de estados é interessante pois consegue prever o caminho que uma autenticação poderá seguir. Em nosso trabalho as interações entre os principais poderá assumir diversos caminhos e um mapeamento para uma máquina de estados traria um formalismo para a nossa proposta.

No trabalho [Foley et al. 2004] é apresentado uma infra-estrutura de segurança para *middlewares* heterogêneos. Para coordenar as relações de confiança entre os diferentes sistemas foi adotado o Keynote [Blaze et al. 1999], porém a infra-estrutura também provê suporte para o SPKI/SDSI .

As políticas de autorização de cada *middleware* são codificadas em certificados do KeyNote e vice-versa. Isso permite que domínios de segurança heterogêneos sejam transpostos, servindo de base para o suporte descentralizado das políticas de segurança. O trabalho detalha as vantagens dos sistemas que usam o gerenciamento de confiança sobre os sistemas que usam o X.509.

Foley tem o objetivo de transpor limites impostos pelas tecnologias através de certificados do Keynote [Foley et al. 2004]. Em nosso modelo propomos ultrapassar tais limites através do uso de padrões para Serviços *Web*, no caso o WS-Trust, que nos parece mais adequado por se tratar de um padrão em caminho de definição. A transposição de limites trouxe problemas para a localização dos direitos necessários para cada domínio e assim descrevemos como contornar tais problemas através do conceito de federações.

6. Conclusões

Através de padrões abertos, os Serviços *Web* surgiram com o intuito de ser uma tecnologia integradora, cobrindo as dificuldades apresentadas pelos modelos anteriores, como a possibilidade de atravessar filtro de pacotes e o uso do XML nas trocas das mensagens.

Descrevemos neste trabalho uma forma para integrar aplicações que utilizem diferentes tecnologias de segurança. Foram utilizadas propostas de segurança para Serviços *Web* juntamente com padrões de segurança para XML para consistir como base de segurança para a arquitetura dos Serviços *Web*, provendo confidencialidade, integridade e autenticidade, e ainda um meio para localização de atributos de segurança, possibilitando assim criar relações de confiança dinâmicas.

Como exemplo, ilustramos dois domínios de segurança, um utilizando o SPKI/SDSI e outro o X.509. Atualmente o modelo X.509 é o mais adotado, porém o

mesmo possui problemas e dificuldades. Já o SPKI/SDSI seria a solução ideal mas não possui ampla aceitação, devido ao fato de ser um modelo relativamente novo e principalmente por que, mesmo com os problemas do X.509, este é um modelo amplamente aceito o qual atende as necessidades básicas atuais. Assim, o trabalho possibilitou uma transição dos modelos de forma gradual, através do uso dos Serviços *Web*.

Nesse trabalho a questão de confidencialidade dos clientes dos serviços não foi abordada. Especificações como a *Liberty Aliance* [Liberty 2003] e a proposta WS-Federation [WS-Federation 2003] propõem serviços de pseudônimos que permitem garantir tal confidencialidade. Como trabalho futuro ficaria assim a adoção e a adequação do uso de serviços de pseudônimos, satisfazendo exigências de um certo nicho de aplicações.

Referências

Ancajima, G. M. C., de Mello, E. R., and da Silva Fraga, J. (2004). Integração da arquitetura de segurança dos serviços web com modelos de confiança igualitária. In *Anais SSI 2004*, São José dos Campos, SP - Brazil. ITA, SSI.

Bartel, M., Boyer, J., and Fox, B. (2002). *XML-Signature Syntax and Processing*. W3C. `http://www.w3.org/TR/xmldsig-core`.

Blaze, M., Feigenbaum, J., Ioannidis, J., and Keromytis, A. (1999). *The keynote trust-management system version 2*. Internet Engineering Task Force RFC 2704.

Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). *Web Services Description Language 1.1*. W3C Working Group.

Ellison, C. M., Frantz, B., Lampson, B., Rivest, R., Thomas, B. M., and Ylonen, T. (1999). *SPKI Certificate Theory*. Internet Engineering Task Force RFC 2693.

Foley, S. N., Quilinan, T. B., O'Connor, M., Mulcahy, B. P., and Morrison, J. P. (2004). A framework for heterogeneous middleware security. In *18th International Parallel and Distributed Processing Symposium (IPDPS'04)*.

Ford, W. and Hallam-Baker, P. (2001). *XML Key Management Specification (XKMS)*. W3C. `http://www.w3.org/TR/xkms`.

Foster, I. and Kesselman, C. (1999). *The grid: blueprint for a new computing infrastructure*, chapter A Toolkit-Based Grid Architecture, pages 259 – 278. Morgan Kaufmann Publishers Inc.

Freier, A. O., Karlton, P., and Kocher, P. C. (1996). *The SSL protocol - v.3*. Internet Draft.

Gnutella (2001). *The Gnutella Protocol Specification v0.4*. Clip2. Doc. rev. 1.2.

Imamura, T., Dillaway, B., and Simon, E. (2002). *XML Encryption Syntax and Processing*. W3C. `http://www.w3.org/TR/xmlenc-core`.

ITU-T (1993). ITU-T recommendation x.509.

Kohl, J. and Neuman, C. (1993). *The Kerberos Network Authentication Service (v5)*. Internet Engineering Task Force RFC 1510.

Liberty (2003). *Liberty Architecture Overview v1.1*. Liberty Alliance.

OASIS (2002). *Security Assertion Markup Language (SAML)*. OASIS. `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security`.

OASIS (2004a). *Universal Description, Discovery and Integration v3.0.2 (UDDI)*. Organization for the Advancement of Structured Information Standards (OASIS).

OASIS (2004b). *Web Services Security: SOAP Message Security 1.0*. OASIS. http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.

OASIS (2005). *eXtensible Access Control Markup Language (XACML) version 2.0*. Organization for the Advancement of Structured Information Standards (OASIS). http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec%-os.pdf.

OMG (2002). Object Management Group - The Common Object Request Broker Architecture v3.0.2. OMG Document 02-12-06.

Santin, A., Fraga, J., Mello, E., and Siqueira, F. (2003). Teias de Federações como extensões ao modelo de autenticação e autorização SDSI/SPKI. In *Anais XXI Simpósio Brasileiro de Redes de Computadores*, pages 553 – 568, Natal, RN - Brazil. SBRC.

Skogsrud, H., Benatallah, B., and Casati, F. (2003). Modelo-driven trust negotiation for web services. In *IEEE Internet Computing*, pages 45– 52. IEE Computer Society.

VeriSign (2002). *VeriSign Digital Trust Services: Enabling Trusted Web Services*. VeriSign.

W3C (2004). *Web Services Architecture*. W3C Working Group.

Winslett, M., Yu, T., Seamons, K. E., Hess, A., Jacobson, J., Jarvis, R., Smith, B., and Yu, L. (2002). Negotiating trust on the web. In *IEEE Internet Computing*, pages 30–37.

Wlech, C., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Meder, S., Pearlman, L., and Tuecke, S. (2003). Security for grid services. In *12th IEEE Int. Symp. on High Performance Distributed Computing*.

WS-Federation (2003). *Web Services Federation Language*. http://msdn.microsoft.com/ws/2003/07/ws-federation.

WS-Policy (2004). *Web Services Policy Framework*. http://msdn.microsoft.com/ws/2004/09/policy/.

WS-PolicyAssertions (2003). *Web Services Policy Assertion Language*. http://msdn.microsoft.com/ws/2002/12/PolicyAssertions.

WS-PolicyAttachment (2004). *Web Services Policy Attachment*. http://msdn.microsoft.com/ws/2004/09/policyattachment.

WS-SecurityPolicy (2005). *Web Services Security Policy Language*.

WS-Trust (2005). *Web Services Trust Language (WS-Trust)*.

Yavatkar, R., Pendarakis, D., and Guerin, R. (2000). *A Framework for Policy-based Admission Control*. IETF RFC 2753.