O uso do SPKI/SDSI em redes P2P

Emerson Ribeiro de Mello[1]*, Joni da Silva Fraga[1] , Altair Olivo Santin[2]

[1]Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
CP 476 – 88040-900 Florianópolis, SC

[2]Programa de Pós-Graduação em Informática Aplicada
Pontifícia Universidade Católica do Paraná
80215901 Curitiba, PR

{emerson,fraga}@das.ufsc.br, altair.santin@pucpr.br

Abstract. *This paper presents a form to integrate trust chains and peer-to-peer networks. The peer-to-peer networks are used to locate SPKI/SDSI certificates chains, which is the main difficult of SPKI/SDSI model. The SPKI/SDSI is used to provide security guarantees as authenticity, confidentiality and a fine access control in peer-to-peer applications. This proposal provides anonymity and reputation properties, which are important for the majority peer-to-peer applications.*

Resumo. *Este trabalho apresenta uma forma de integrar redes de confiança com redes par a par. Em nossa proposta utilizamos as redes par a par para solucionar a principal dificuldade do modelo SPKI/SDSI, que é a localização das cadeias de confiança. Já o SPKI/SDSI é utilizado para prover uma maior segurança para aplicações que usufruam de redes par a par, garantindo a autenticidade, integridade, confidencialidade e um controle de acesso de fina granularidade. A solução proposta ainda se preocupa com as propriedades de anonimato e reputação, propriedades que são importantes para as mais populares aplicações par a par.*

1. Introdução

Atualmente a maioria das aplicações presentes na Internet segue o modelo cliente-servidor, onde uma máquina, que possui um endereço conhecido e raramente modificado, provê serviços para outras máquinas, denominadas clientes. Este modelo concentra normalmente todos os recursos disponibilizados em uma única máquina (o servidor). Com uma maneira diferente de operar, as redes par a par, também conhecidas como redes P2P (*peer-to-peer*), provêem serviços de uma forma descentralizada, onde cada nó pertencente a rede poderá buscar por serviços, como também provê-los.

Nas redes P2P os conceitos básicos de segurança [7] como a integridade, confidencialidade, autenticidade e autorização geralmente não são tratados com o intuito de evitar o tempo extra para o processamento e transmissão (*overhead*) [10]. As aplicações P2P destinadas a troca de conteúdos, como músicas, livros eletrônicos, são as menos providas de segurança. Em algumas aplicações há preocupação com a autorização porém, existe uma total ausência de preocupação com relação à autenticação.

*Bolsista CNPq.

Para o ambiente P2P seria interessante uma infra-estrutura de chaves públicas que pudesse garantir a autenticidade, que tivesse um controle de acesso granular e confiabilidade sobre as informações trocadas e que ainda garantisse o anonimato dos usuários. Certificados X.509 [6] formam uma base para garantir a autenticidade. Porém, as redes P2P são dinâmicas, os nós entram e saem da rede de uma maneira rápida e nem sempre entram com o mesmo IP, requisito básico para a geração dos certificados. Fato o qual não é um problema para o modelo SPKI/SDSI (*Simple Public Key Infrastructure / Simple Distributed Security Infrastructure*) [[4], [11]], visto que o modelo não se baseia em topologias rígidas de confiança, como o X.509.

O SPKI/SDSI é uma infra-estrutura de chaves públicas completamente descentralizada e é voltada para a autorização e não para a autenticação. No SPKI/SDSI cada principal[1] é caracterizado por um par de chaves, estando apto a emitir e assinar certificados, não dependendo assim de uma terceira parte confiável, como no caso do X.509 que depende de uma Autoridade Certificadora (AC). Baseado no modelo de *redes de confiança* [1], os certificados SPKI/SDSI podem ser delegados de principais para principais. Para isto, é necessário que exista uma relação de confiança entre os mesmos. As inúmeras delegações de certificados formam as cadeias de certificados, cadeias as quais deverão ser apresentadas no momento de acesso ao recurso. Na especificação do SPKI/SDSI não é definido como realizar a busca pelas cadeias de certificados, sendo esta a principal dificuldade do modelo. Já foram feitas diversas propostas para solucionar o problema, porém nestas propostas sempre há um certa centralização do serviço, algo que difere do modelo SPKI/SDSI. Assim, a proposta apresentada neste artigo, busca utilizar as redes P2P como um meio para ajudar na busca de certificados e onde o modelo SPKI/SDSI propiciará a segurança necessária para as redes P2P.

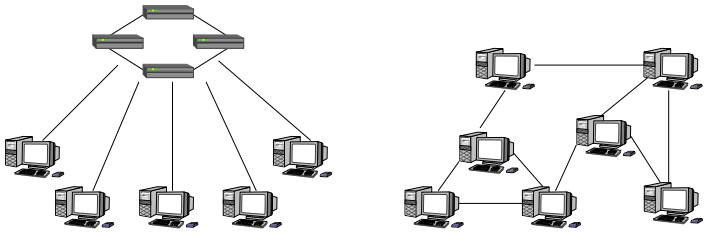
A estrutura deste artigo está definida da seguinte forma. Na seção 2 é descrito o conceito de redes P2P. O modelo de segurança SPKI/SDSI é mostrado na seção 3. Na seção 4 é apresentado o modelo proposto por este trabalho. Na seção 5 estão os trabalhos relacionados e na seção 6 está a conclusão.

2. Redes par a par

As redes par a par são caracterizadas pela variação constante e rápida do número de nós que fazem parte da rede. Trata-se de um modelo igualitário, onde todos os nós participantes possuem as mesmas habilidades. Assim, cada nó é capaz tanto de distribuir a informação bem como obtê-la. Redes P2P acessam recursos descentralizados através de um sistema com conectividade instável, sem possuir dependência do serviço de nomes (*Domain Name System* – DNS) [8] e com a autonomia total ou quase total, em relação a servidores centralizados.

As redes P2P podem seguir basicamente dois modelos: ***redes híbridas*** e ***redes puras***. As *redes híbridas* (figura 1(a)) são caracterizadas pelo fato de possuírem uma pequena parte centralizada que pode ser, por exemplo, um catálogo para listar os recursos disponíveis. Já as *redes P2P puras* (figura 1(b)) são completamente descentralizadas, onde somente uma pequena parte da inicialização necessita de um lugar conhecido para que se possa buscar uma lista contendo um subconjunto de todos os nós presentes na rede naquele instante.

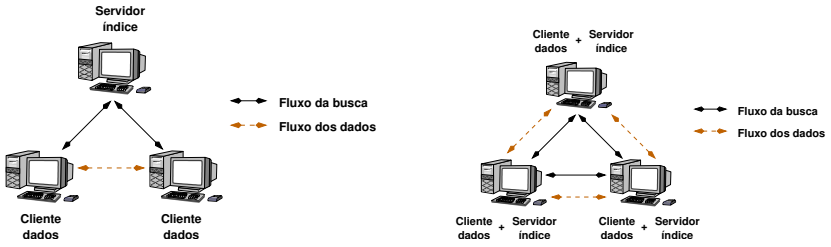
[1] Usuário, processo ou máquina autorizados pelas políticas do sistema.



(a) Híbrida (b) Pura

Figura 1. Topologias das redes P2P

As **redes híbridas** operam com um nó central (ou um conjunto de nós) que mantém um índice sobre os demais nós ativos na rede e sobre os recursos disponibilizados por estes. Uma vez que se tenha localizado o recurso desejado, a comunicação entre o nó requerente e o nó provedor de recurso é realizada diretamente, não possuindo qualquer dependência com o nó central (veja a figura 2(a)). O Napster e ICQ são exemplos aplicações que utilizam redes P2P híbridas. O Napster foi a primeira aplicação do gênero para troca de músicas pela Internet. Contava com um servidor centralizado que disponibilizava todos os conteúdos compartilhados por todos os nós da rede e assim uma forma para localizar os nós detentores de tais conteúdos. Porém, uma vez que o conteúdo desejado é encontrado, a comunicação entre os nós (cliente e provedor do conteúdo) é realizada diretamente, sem o intermédio dos servidores do Napster (veja a figura 2(a)). A rede *FastTrack*, utilizada pelos aplicativos Kazaa e Morpheus, segue a topologia "centralizada + descentralizada", onde muitos nós possuem um relacionamento centralizado com um super nó, encaminhando a este todas suas as requisições. Porém, o relacionamento entre os super nós é descentralizado.



(a) Índice centralizado (b) Índice e conteúdo descentralizado

Figura 2. Disposição do índice de recursos

As **redes P2P puras** operam sem a necessidade de um nó central. Cada nó da rede possui um índice parcial, o qual representa um subconjunto de todos os nós participantes da rede. Um nó, ao realizar uma busca por um conteúdo, questionará todos os nós presentes em seu índice parcial afim de obter uma resposta. Cada nó ao receber a consulta, verificará se possui o conteúdo desejado e caso o tenha, responderá para o nó que lhe questionou. Caso contrário irá propagar a busca para os nós presentes em seu índice parcial. Ao encontrar o conteúdo desejado, é criado um caminho reverso à consulta, onde o

nó que possui o conteúdo irá responder ao nó que lhe questionou e assim sucessivamente até a resposta chegar ao nó que originou a consulta.

Uma vez na rede, o nó fica completamente independente de um local conhecido para obter o índice com os nós ativos e este índice estará em constante atualização utilizando a própria infra-estrutura da rede P2P. O Gnutella [2] é um exemplo de uma aplicação P2P pura (veja a figura 2(b)).

2.1. Segurança em Redes P2P

Associados aos benefícios trazidos pelas redes P2P, estão alguns problemas. Informações sobre os recursos disponibilizados podem estar incorretas ou os recursos disponibilizados podem estar corrompidos. Por exemplo, uma música poderia ser listada como sendo de um determinado artista e na realidade ser de outro. Ou ainda, o arquivo pode não ser a música desejada mas sim um arquivo malicioso. Outra dificuldade é a dinâmica da rede, informações desejadas e listadas podem não mais estarem disponíveis devido ao fato do nó já ter saído da rede.

Atualmente as principais aplicações P2P para troca de conteúdo não se preocupam com os princípios básicos de segurança, integridade, autenticidade, confidencialidade e ainda a autorização. Por exemplo, o ICQ garante a autenticidade dos usuários através de um servidor de autenticação centralizado, porém o protocolo do ICQ não trata a integridade e a confidencialidade. Algumas implementações de terceiros, que utilizam o protocolo do ICQ, como o LICQ e o SIM, utilizam o SSL [5] para garantir a confidencialidade fim a fim. Com relação ao controle de acesso, é possível fazer uma analogia entre as listas de "*visíveis*" e "*invisíveis*" do ICQ com o modelo RBAC CORE [12], onde o dono da lista só aparecerá visível para os contatos que estiverem contidos na lista de *visíveis* e nunca aparecerá visível para os contatos que estiverem presentes na lista de *invisíveis*.

Assim, é possível afirmar que o ICQ consegue cobrir os requisitos básicos de segurança, porém a grande maioria das aplicações P2P para troca de conteúdo não estão tão bem providas assim. Algumas implementações permitem classificar os recursos compartilhados e somente disponibilizá-los para usuários que possuam um papel que lhe garanta o acesso. Porém, há dificuldade em garantir a autenticidade dos usuários sem a necessidade de uma terceira parte confiável, já que as listas de controle de acesso são baseadas na identidade do usuário. Nas atuais aplicações P2P para o compartilhamento de conteúdos, cada usuário possui uma identificação única na rede, porém não possui uma autoridade certificadora para garantir tais identidades.

3. SPKI/SDSI

A especificação do SPKI/SDSI (*Simple Public Key Infrastructure / Simple Distributed Security Infrastructure*) [[4], [11]] descreve uma infra-estrutura de chaves públicas que permite que a autenticação e a autorização sejam realizadas de forma idependente de qualquer ponto centralizador, fato que não ocorre com o X.509, onde a autenticação é garantida através de uma terceira parte confiável que ambos, cliente e servidor, devem previamente conhecer e confiar.

O SPKI/SDSI usa um modelo igualitário. Os principais são chaves públicas e cada chave pública é uma autoridade certificadora que pode assinar e divulgar certificados. Por não ser uma infra-estrutura de hierarquia global, as comunidades SPKI/SDSI são construídas na forma *bottom-up* e não requerem uma raiz confiável.

Existem dois tipos de certificados no SPKI/SDSI: certificados de autorização – que delegam direitos de um principal para outro; e certificados de nome – que atribuem um nome a uma chave pública ou a um outro certificado de nome. Os problemas relacionados com sistemas de nomes globais são evitados no SPKI através das cadeias de certificação. Essas cadeias são formadas por chaves públicas ligadas entre si, delegando autoridade de uma para outra. A principal característica adicionada pelo SDSI foi a noção de espaço de nomes local. Desta forma é possível definir um nome a um grupo, referenciar todos os integrantes do grupo por este nome (figura 3(a)) e emitir um único certificado de autorização para o grupo (figura 3(b)). Caso alguma alteração na autorização seja necessária, basta fazê-la para o nome do grupo e esta será herdada por todos os seus integrantes.

(a) Emissão de um certificado de grupo

(b) Emissão de um certificado de autorização

Figura 3. Delegação de direitos para um certificado de grupo

Baseado no conceito de *Redes de confiança* [1], onde cada principal determina em quais principais confiar, a busca pelas cadeias de certificados é uma tarefa difícil por não existir um repositório conhecido que contenha todos os certificados necessários para formar a cadeia desejada.

4. O uso conjunto do SPKI/SDSI com as redes par a par

A proposta busca utilizar o modelo SPKI/SDSI juntamente com as redes par a par afim de cumprir as propriedades de rede arbitrárias e dinâmicas dos dois modelos. Nesta seção serão apresentados primeiramente os benefícios adquiridos para o modelo SPKI/SDSI com o uso das redes par a par e depois as vantagens em utilizar o modelo SPKI/SDSI em aplicações par a par destinadas ao compartilhamento de conteúdos digitais.

4.1. Redes P2P SPKI - A busca pelas cadeias de certificados

O SPKI/SDSI pode ser utilizado em qualquer tipo de aplicação que necessite garantir propriedades de autenticidade e controle de acesso. O SPKI/SDSI surgiu como uma alternativa para o X.509 mas ainda não teve sua adoção consolidada em uma implementação de sucesso, como o SSL para o X.509. A figura 4 ilustra um cenário cliente/servidor típico, onde *N1* (o cliente) deseja acessar um recurso (protegido) de *N2*. A figura ainda ilustra uma rede P2P pura, denominada P2P SPKI, onde cada nó possui uma aplicação P2P SPKI que tem por objetivo auxiliar na localização de certificados.

A especificação do SPKI/SDSI não prevê uma forma para localização destes certificados. Para suprir essa dificuldade, lançamos uma heurística de busca (seção 4.3.) baseada no protocolo Gnutella, conseguindo assim obter escala sem que haja necessidade de um nó centralizado na rede onde todos os outros demais nós ativos estejam conectados.

Na figura 4, as linhas contínuas indicam a conectividade entre os nós da rede P2P, informando para o nó quais nós vizinhos estão ativos. As linhas tracejadas indicam as relações de confiança entre nós, onde tais relações de confiança não implicam em uma conectividade entre os nós na rede P2P. A aplicação permite que sejam feitas buscas por cadeias de certificados, informando: emissor, sujeito, nome ou direito (*tag*), se é delegável e o período de validade.

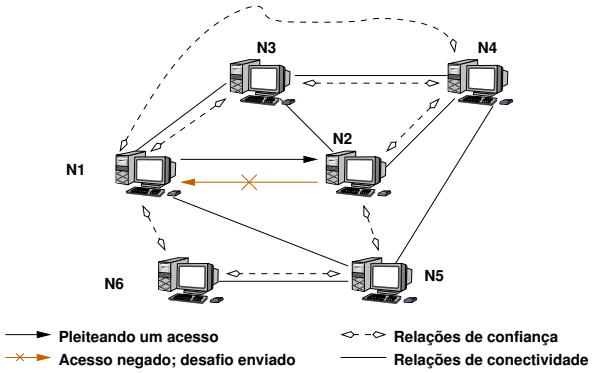


Figura 4. Rede P2P SPKI

No cenário de exemplo, ilustrado pela figura 4, um cliente (nó *N1*) deseja acessar um recurso provido pelo nó *N2*. O mecanismo de controle de acesso presente em *N2* verifica se o pedido enviado por *N1* contém os direitos de acesso necessários para garantir o acesso ao recurso. Os direitos poderiam ser um certificado ou cadeias de certificados. Se o pedido, enviado por *N1*, não contiver os direitos requeridos, então *N2* envia para *N1* uma lista com os principais que possuem tais direitos e que possam delegá-los[2].

Supondo que *N1* ao realizar uma busca pelos possíveis detentores dos direitos necessários para o acesso ao *N2*, obtenha como resposta os principais *N4* e *N5*. Então, a aplicação presente em *N1*, decide com qual destes principais deseja negociar um certificado. Para a decisão, a aplicação de *N1* prioriza os principais com quem *N1* já possui alguma relação de confiança prévia e por fim tenta negociar com os demais principais.

O modelo SPKI/SDSI não descreve como deverá ser realizada a negociação de direitos, deixando livre para os desenvolvedores da aplicação. Supondo que, *N1* ao realizar a busca para encontrar os principais que possuem os direitos requeridos por *N2*, encontre somente o principal *N5* e se a negociação entre *N1* e *N5* ocorrer com sucesso, basta *N1* apresentar os direitos obtidos de *N5* ao *N2* e assim o acesso ao recurso estará garantido. Porém, caso não haja sucesso na negociação, a rede P2P SPKI pode ser utilizada para a localização de caminhos de confiança alternativos. Dessa forma, uma nova busca será realizada, procurando por principais que possuam "alguma relação de confiança"[3] com *N5*, e não mais com *N2*. A busca por principais comuns entre *N1* e *N5* aumenta a possibilidade da concessão dos direitos desejados, sendo ainda uma solução tolerante a faltas, já que os direitos poderão ser delegados por quaisquer principais intermediários.

[2]O funcionamento da política de negócios é de livre escolha do desenvolvedor da aplicação, *N2* ao invés de lançar um desafio informando onde obter os direitos, poderia negociar a venda destes direitos.

[3]Uma relação de confiança pode ser caracterizada pela existência de algum certificado de autorização que tenha sido delegado por *N5*.

No cenário apresentado na figura 4, a busca por principais comuns a *N1* e *N5* obteve como resultado somente o principal *N6*. Uma possível política de negócios presente na aplicação do nó *N1*, ao verificar que a negociação direta entre *N1* e *N5* não obteve sucesso, poderia então buscar por uma solução alternativa e assim invocar o principal *N6*. Existem duas possibilidades para *N1* questionar *N6*. Uma possibilidade seria *N1* solicitar a *N6* um certificado de nome, indicando que ele (*N6*) "conhece" *N1* e possui algum tipo de relação de confiança estabelecida. Assim, *N1* apresenta tal certificado para *N5* afim de que o mesmo possa agora conceder os direitos desejados. A política de negócios de *N5* pode definir que confia nos principais, com quem já possui alguma relação de confiança, e também nos "amigos" destes. Desta forma, *N5* delegaria os direitos desejados por *N1*, já que *N1* provou ser "amigo" de um principal que *N5* confia. Para o caso onde a política de negócios de *N5* fosse mais rigorosa, informando que só delega direitos para os principais com quem possui alguma relação de confiança, uma possibilidade para *N1* seria: *N6* solicita os direitos, requeridos por *N2*, ao principal *N5* e uma vez que tiver os direitos em mãos, *N6* poderia delegá-los para *N1*.

4.2. Aprimorando a segurança nas aplicações P2P

Existem diversos tipos de aplicações que usufruem das redes P2P. Sejam aplicações para troca de mensagens instantâneas, para armazenamento de arquivos remotos ou ainda para o compartilhamento de conteúdos digitais. Para cada tipo de aplicação pode-se desejar um certo nível de segurança. Por exemplo, nas trocas de mensagens instantâneas é importante garantir a autenticidade das partes, já em aplicações que compartilham arquivos MP3, é interessante garantir o anonimato. Um ponto comum entre todas aplicações P2P é a necessidade do controle de acesso onde o mesmo evitaria que arquivos confidenciais fossem compartilhados com pessoas não autorizadas.

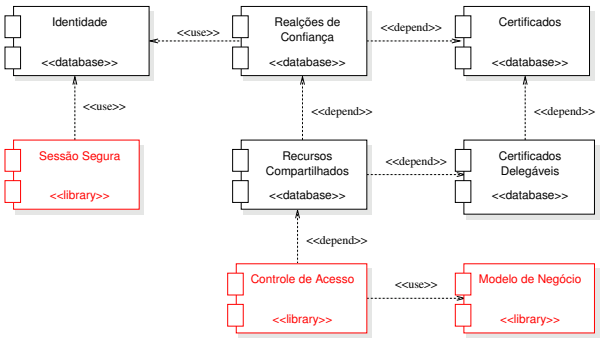


Figura 5. Componentes da aplicação P2P SPKI

Serão apresentadas aqui formas para cobrir todas as propriedades básicas de segurança bem como as propriedades de anonimato e reputação dos nós P2P. Esta base poderá ser adotada de forma completa ou parcial em qualquer tipo de aplicação P2P. A figura 5 ilustra a arquitetura de segurança proposta para a aplicação P2P SPKI e a descrição sobre cada componente é mostrada a seguir, tendo como exemplo uma aplicação P2P para compartilhamento de conteúdos digitais.

Geralmente, para um nó ingressar em uma rede P2P é necessário que o mesmo o faça apresentando uma identidade única na rede. Em nossa proposta, essa identificação é feita através de um par de chaves SPKI/SDSI. Como comentado na seção 3., no

SPKI/SDSI os principais são identificados como chaves públicas, o que garante o anonimato dos usuários, anonimato o qual poderia ser ferido se o endereço de e-mail fosse utilizado na identificação. Cada usuário pode possuir a quantidade de identidades que desejar, sabendo que os certificados de autorização que este receber ou emitir estarão diretamente relacionados com cada par de chaves.

As mais populares aplicações P2P possuem somente um repositório para armazenar os recursos que serão compartilhados e o controle de acesso sobre tais recursos está baseado em uma implementação do modelo RBAC, onde recursos e usuários são associados com papéis. Assim, os usuários só poderão obter os recursos os quais seu papel permitir. Geralmente os papéis são definidos estaticamente, impossibilitando que cada usuário faça seus próprios ajustes de acordo com suas necessidades.

A aplicação P2P SPKI proposta, possui quatro tipos de repositórios. O *Repositório de Certificados* – armazena certificados de autorização que foram delegados para o usuário em questão; *Repositório de Certificados Delegáveis* – armazena somente os certificados de autorização recebidos e que podem ser delegados para outros principais, podemos dizer que é um subconjunto do *Repositório de Certificados*; *Repositório de Relações de Confiança e de Reputações* – armazena certificados de autorização e também mantém uma lista de reputações sobre todos os principais com quem o usuário já realizou algum tipo de comunicação, por exemplo, a troca de um arquivo; e por fim o *Repositório de Recursos Compartilhados* – armazena todos os recursos que o usuário deseja compartilhar com os demais nós da rede P2P, inclusive os certificados de autorização que podem ser delegáveis, relações de confiança, bem como a tabela de reputações que o usuário achar interessante compartilhar.

Aplicações P2P possuem mecanismos para determinar se um arquivo está corrompido, como por exemplo, um arquivo danificado mas mesmo assim é comum encontrar arquivos corrompidos. Isto ocorre visto que as aplicações não preocupam-se em garantir se o arquivo é realmente o que diz ser. Usuários mal intencionados poderiam disponibilizar arquivos maliciosos com nomes de arquivos populares, fazendo com muitos usuários sejam enganados. Algumas redes P2P possuem um sistema onde os próprios usuários opinam sobre a qualidade do arquivo que possuem. Trata-se de um repositório centralizado de "opniões sobre os arquivos", alimentado por qualquer usuário da rede. Assim, antes que um usuário opte em obter um arquivo desejado, é possível verificar se o arquivo possue alguma classificação positiva ou negativa e assim decidir.

Esse tipo de sistema funciona relativamente bem, porém não possui garantias contra usuários mal intencionados. Alguns usuários poderiam entrar em acordo para dar pareceres positivos sobre determinado arquivo, mesmo sabendo que trata-se de um arquivo malicioso. Esse problema ocorre porque os usuários só confiam em uma fonte centralizada, cuja base de dados é administrada por usuários da própria rede P2P, mesmo que estes não sejam confiáveis.

Em nossa arquitetura o sistema de reputação está diretamente ligado aos certificados de autorização SPKI onde cada usuário da rede informa em quem pode-se confiar. Um usuário *U1* poderia dizer que todas as informações provenientes do usuário *U2* são confiáveis. Ou ainda, um usuário *U1* poderia dizer que todas informações provenientes do usuário *U2*, e de todos os seus amigos, são confiáveis, formando assim as *Redes de Confiança*.

Em um outro caso, após o sucesso de uma negociação com usuário que é dito *não confiável*, por exemplo *U3*, este seria adicionado na tabela de reputação com um valor de referência padrão, por exemplo 100. Os usuários com os maiores valores na tabela são apontados como os mais confiáveis, visto que experiências anteriores e futuras determinam suas posições. *U1* pode ainda emitir um certificado de autorização para *U3* indicando, por exemplo, que realizou *N* operações com *U3* sem que tenha obtido qualquer arquivo corrompido. Tal certificado é considerado como uma avaliação positiva e *U3* poderia informar as avaliações que recebeu como forma de provar para outros usuários que ele, *U3*, honrou comunicações anteriores com outros usuários[4].

A biblioteca de *Controle de Acesso* de nossa proposta está baseada no modelo RBAC juntamente com a filosofia de certificados do SPKI/SDSI. Cada usuário pode emitir papéis (certificados de nomes) para seus amigos e assim associar tais papéis com os recursos. Essa flexibilidade permite adicionar ou remover usuários ou recursos de um papel sem grandes esforços, lembrando ainda que os papéis podem possuir um período de validade. Porém, a maior facilidade dada pelo SPKI/SDSI é delegação de direitos. Um usuário *U1* poderia emitir um certificado de autorização para um usuário *U2* permitindo que o mesmo acesse todos os recursos que possuam o papel "amigos" e ainda, que o mesmo possa delegar tal direito para outros usuários. Assim, se *U2* delegar o certificado para *U3*, todos os recursos providos por *U1* também estarão acessíveis para *U3*.

Na arquitetura é apresentada a biblioteca "Modelo de negócio", cuja biblioteca "Controle de Acesso" depende. A idéia básica do controle de acesso é verificar se o requerente possui ou não os direitos necessários para obter o recurso desejado. Nos mecanismos de controle de acesso tradicionais o simples fato de não fornecer os direitos iniciais requeridos, indica o término da negociação. Caso o requerente não detenha os direitos necessários, é possível que o mesmo forneça algum outro tipo de informação que possa ser interessante para o provedor do recurso. Por exemplo, *N1* deseja um recurso compartilhado por *N5*, porém, não possui os direitos necessários para isso. A política de negócios em *N5* poderia informar a *N1* para que ele forneça alguns certificados de autorização que possui e que são de interesse de *N5*. Assim, neste caso houve uma troca de favores. *N1* fornece alguns certificados de autorização que possui para *N5* e em troca *N5* concede o acesso ao recurso desejado.

Por fim, a biblioteca "Estabelecimento de Sessão Segura" provê meios para garantir as propriedades de confidencialidade e integridade. Qualquer uma das partes envolvida em uma comunicação pode requisitar um canal seguro e neste caso é usado um canal SSL.

4.3. Heurística da busca por direitos

Como infra-estrutura de suporte, para a busca dos direitos ou relações de confiança, foi utilizado o protocolo Gnutella. A *busca* é enviada para todos os vizinhos do nó P2P (nós que possuem conectividade entre si) e a *resposta* volta pelo caminho inverso até chegar ao emissor da *busca*. A partir deste ponto a transferência ocorre diretamente entre o emissor e aquele nó que respondeu a *busca*. No caso apresentado na figura 4, *N6* informa que possui relação de confiança com *N5*, em resposta à busca realizada por *N1*. Assim, *N1* e *N6* realizam uma comunicação direta.

[4]Cabe a política de negócios de cada usuário decidir se confia ou não nessas avaliações, onde o caso mais simples seria confiar no principal que emitiu o certificado.

O protocolo que segue este modelo possui duas mensagens: a mensagem *query*, que é usada para efetuar a busca por recursos; e a mensagem *queryHit*, informando que o recurso procurado foi encontrado. O algoritmo 1 descreve como a busca por direitos (ou relações de confiança) é realizada.

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** $(recurso \subset D) OR (recurso \subset T)$ **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: $x \longleftarrow getElement(N)$
8: $P \longleftarrow origem \cap P$
9: $query(noAtual, recurso, P, ttl - 1)$
10: $N \longleftarrow N \setminus \{x\}$ {Remove o elemento x do conjunto N}
11: **end while**
12: **end if**
13: **end if**

A mensagem *query* é composta por quatro variáveis: *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), ao receber a mensagem "*query*" verifica em seus repositórios locais – conjuntos D e T do algoritmo 1 – se o mesmo possui o "*recurso*" procurado 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 – conjunto T – com quem possui relação de confiança (linha 6 – 11). Para cada novo nível que a mensagem "*query*" descer, o valor da variável "*ttl*" é decrementado, evitando que a mensagem se propague indefinidamente.

Na linha 1 do algoritmo 1 é verificado se o *recurso* procurado está contido no conjunto D ou no conjunto T. Isto se justifica, visto que o *recurso* procurado pode ser, por exemplo, um arquivo de música ou um certificado de autorização, conteúdos os quais estariam presentes no conjunto D, ou ainda pode-se desejar saber quem possui relação de confiança com o *principal X*, estando esta informação disponível no conjunto T.

Com esta heurística é possível cobrir uma grande variedade de nós em busca de direitos sem que haja a necessidade de um repositório central, informando quem possui direito sobre o quê. O desempenho do algoritmo pode ser melhorado através da implementação de índices locais, os quais relacionam os recursos providos por cada nó. A manutenção do índice é realizada através das próprias consultas direcionadas ao nó, deixando ali armazenadas as n últimas consultas, por exemplo.

5. Trabalhos relacionados

O trabalho de Cornelli [3] se preocupa em garantir a reputação dos nós de uma rede P2P. Para não ferir o anonimato, Cornelli utiliza o campo *servent_id* presente nas aplicações P2P para identificar o nó na rede, porém, sem relacionar tal informação diretamente com

o usuário. Após um nó *p* receber as respostas dos provedores que possuem o recurso desejado, o nó *p* além de verificar a qualidade dos provedores (largura de banda, posição na fila de espera, etc.), lança uma consulta para seus nós vizinhos, perguntando o que cada um acha sobre esses provedores e com base nas respostas, decide em quem deve confiar ou não. As respostas sobre a reputação podem ser anônimas ou identificadas através do *servent_id*. O principal problema apresentado no trabalho de Cornelli está na personificação dos nós que emitem os "votos sobre a reputação". Assim, Cornelli propõe o uso de chaves públicas para garantir a integridade e a autenticidade dos votos enviados pela rede P2P, tendo como identificador (*servent_id*) um resumo da chave pública do nó.

A solução proposta em nosso trabalho está baseada no uso de chaves e certificados SPKI/SDSI. A solução garante o anonimato dos usuários sem que esteja suscetível a ataques de personificação. Já a propriedade de reputação dos provedores de recursos é melhor tratada. Como no trabalho de Cornelli [3], a reputação dos provedores de recursos pode ser dada através de votos dos nós vizinhos e atribuir pesos para cada voto, de acordo com a confiança que possui com cada nó. A nossa proposta também garante a reputação através dos certificados de autorização concedidos para o provedor do recurso e que são considerados como um aval positivo sobre as experiências anteriores com outros nós. Isso garante uma maior flexibilidade e um melhor desempenho, visto que o nó, que realizou a busca inicial por um recurso, não precisará realizar uma nova busca para consultar a reputação dos provedores.

Em [9] é proposta uma solução para a localização das cadeias de certificados SPKI/SDSI. A proposta faz uso do DNS [8] e de suas extensões de segurança propostas pela RFC 2065. Assim sendo, propõe um certo nível de centralização, visto que os servidores DNS, juntamente com as devidas extensões propostas por Nikander [9], serão responsáveis pela localização das cadeias de certificados.

No trabalho de Santin [13] a solução proposta para localização das cadeias de certificados se baseia no conceito de federações. As Federações SPKI, são compostas por gerentes e por um conjunto de principais que possuem afinidades. Para se obter escala os gerentes das federações criam associações entre si, o que permite que um principal de uma federação consiga localizar uma cadeia de certificados que esteja em posse de um principal de outra federação.

Em ambos trabalhos apresentados acima, existe uma certa centralização do serviço para busca de cadeias de certificados. Em nosso trabalho optamos por uma rede P2P pura, onde cada nó da rede assume os papéis de cliente e servidor. A busca pelas cadeias de certificados é lançada para os nós vizinhos e esses encaminham para os seus vizinhos da mesma forma que funciona o protocolo Gnutella, evitando assim depender de qualquer ponto centralizado. Como nos trabalhos, [9] e [13], a nossa solução visa a localização das cadeias de certificados, porém, como esses certificados serão negociados, fica sob a responsabilidade das aplicações em questão.

6. Conclusão

Este trabalho apresentou uma forma para integrar os modelos SPKI e redes P2P, propiciando uma solução para as dificuldades presentes em cada modelo. Mostramos aqui que é possível utilizar as atuais implementações P2P, sem grandes custos de implementação, para a localização das cadeias de certificados SPKI/SDSI, sendo essa uma solução com-

pletamente distribuída, evitando a centralização parcial apresentada em outros trabalhos.

A contribuição para aplicações P2P foi descrita através de um exemplo de uma aplicação para distribuição de conteúdos digitais, ilustrando ainda uma possível política de negócio para ser utilizada nessas aplicações. O uso do SPKI/SDSI permitiu garantir as propriedades básicas de segurança como a (i) *confidencialidade*, onde, se ambas partes desejarem, é possível utilizar um canal cifrado; (ii) *autenticidade*, garantindo a identidade dos principais de uma rede P2P; (iii) *integridade*, evitando que usuários maliciosos modifiquem mensagens em trânsito; e ainda garantimos propriedades desejadas para aplicações P2P como o (iv) anonimato, apesar dos principais possuírem uma identificação única rede, essa não possibilita identificar o usuário por trás deste principal; e por fim a (v) *reputação*, evitando que usuários maliciosos abusem das redes P2P para a propagação de arquivos maliciosos ou mesmo a propagação de arquivos que não condizem com sua descrição.

A arquitetura de segurança ilustrada na figura 5 serve de base para qualquer tipo de aplicação P2P que queira utilizar certificados SPKI/SDSI e assim obter as propriedades de segurança citadas acima. A arquitetura apresentada já foi bem definida e um protótipo de uma aplicação P2P que usufrui da mesma está em desenvolvimento. O protótipo tem como base o protocolo Gnutella e está sendo desenvolvido em Java. Como trabalho futuro fica o término do protótipo para que se possa obter uma análise dos custos envolvidos, tanto com uso da infra-estrutura de chave pública quanto com as trocas de mensagens envolvidas para a obtenção das cadeias de certificados.

Referências

[1] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. Technical Report 96-17, AT&T, 28, 1996.

[2] Clip2. *The Gnutella Protocol Specification v0.4*, 2001. Doc. rev. 1.2.

[3] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Choosing reputable servents in a p2p network. In *11th international conference on WWW*, pages 376–386. ACM Press, 2002.

[4] C. M. Ellison, B. Frantz, B. Lampson, R. Rivest, B. M. Thomas, and T. Ylonen. *SPKI Certificate Theory*. IETF RFC 2693, Setembro 1999.

[5] A. O. Freier, P. Karlton, and P. C. Kocher. *The SSL protocol - version 3*. Internet Draft, Março 1996.

[6] ITU-T. ITU-T recommendation x.509, 1993.

[7] C. E. Landwehr. Computer Security. In *International Journal of Information Security*, volume 1, pages 3–13. Springer-Verlag Heidelberg, Julho 2001.

[8] P. Mockapetris. *Domain names – Concepts and Facilities*. IETF RFC 1034, 1987.

[9] P. Nikander and L. Viljanen. Storing and Retrieving Internet Certificates. In *Proceedings of the Third Nordic Workshop on Secure IT Systems*, 1998.

[10] M. Parameswaran, A. Susarla, and A. B. Whinston. P2P networking: An information-sharing alternative. *IEEE Computer*, pages 31–38, Julho 2001.

[11] R. L. Rivest and B. Lampson. SDSI – A simple distributed security infrastructure. Presented at CRYPTO'96 Rumpsession, 1996.

[12] R. S. Sandhu and P. Samarati. Access control: Principles and practice. *IEEE Communications Magazine*, 32(9):40–48, 1994.

[13] A. Santin, J. Fraga, E. Mello, and F. Siqueira. 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, 2003. SBRC.