UNISINOS

Programa Interdisciplinar de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

Janaína Conceição Sutil Lemos

Sec-SD: Um Modelo Distribuído para Descoberta de Serviços em Redes Locais

São Leopoldo, 2011

Janaína Conceição Sutil Lemos

SEC-SD: UM MODELO DISTRIBUÍDO PARA DESCOBERTA SEGURA DE SERVIÇOS EM REDES LOCAIS

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Rafael Bohrer Ávila

São Leopoldo
2011

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

Lemos, Janaína Conceição Sutil

Sec-SD: Um Modelo Distribuído para Descoberta Segura de Serviços em Redes Locais / Janaína Conceição Sutil Lemos — 2011.

96 f.: il.; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2011.

"Orientador: Prof. Dr. Rafael Bohrer Ávila, Unidade Acadêmica de Pesquisa e Pós-Graduação".

1. Sistemas distribuídos. 2. Descoberta de serviços. 3. Segurança. 4. Criptografia. 5. Autenticação. I. Título.

CDU 004.732

Bibliotecária responsável: Vanessa Borges Nunes — CRB 72.056

JANAÍNA CONCEIÇÃO SUTIL LEMOS

Dissertação apresentada como requisito parcial para a obtenção título de mestre, pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos.

Aprovado em 30 de Setembro de 2011

BANCA EXAMINADORA

Orientador: Prof. Dr. Rafael Bohrer Ávila – Unisinos

(Examinador interno) Prof. Dr. Cristiano André da Costa – Unisinos

---

(Examinadora externa) Profª. Drª. Márcia Pasin – UFSM

Aos meus pais.

AGRADECIMENTOS

Agradeço a Deus por ter chegado até aqui.
Aos meus pais Lindomar (em memória) e Vera, pela educação e por terem sempre acreditado em mim.
A Dalva, que também é uma mãe para mim, por nunca ter deixado eu desistir.
Ao meu orientador, Prof. Rafael Bohrer Ávila, pela sua orientação, dedicação, incentivo e por estar sempre disposto a ouvir.
Aos demais professores do PIPCA, em especial ao Prof. Luiz Paulo Luna de Oliveira, pelas suas contribuições e pelo incentivo.
A todos os amigos que fiz aqui, em especial ao Érico Santos Rocha, pelo incentivo e pelas críticas construtivas :)
E por fim, caso eu tenha esquecido de alguém, quero agradecer a todos que de alguma forma contribuíram para que eu chegasse até aqui.

"*A neve e as tempestades matam as flores, mas nada podem contra as sementes*".
(Khalil Gibran)

RESUMO

Com a crescente popularização dos dispositivos móveis nos últimos anos, há uma necessidade cada vez maior de conectividade e de serviços nas redes de computadores. Nesse contexto, as tecnologias para descoberta de serviços simplificam a interação entre usuários e dispositivos, facilitando as tarefas administrativas, principalmente quando existe a necessidade de adicionar novos equipamentos. Devido a grande diversidade de ambientes onde essas tecnologias podem ser utilizadas, surge também a necessidade de tratar as questões relacionadas a segurança e ao mesmo tempo, preservar a facilidade de uso do sistema. Neste trabalho é apresentado um sistema para descoberta segura de serviços em redes locais com arquitetura descentralizada, o Sec-SD (*Secure Service Discovery Protocol*). Através do uso de mecanismos para criptografia e autenticação, o Sec-SD visa estabelecer uma relação de confiança entre as partes envolvidas na descoberta de serviços antes da divulgação de informações relacionadas a estas, prevenindo assim os ataques causados pelo anúncio de falsos serviços, bem como o acesso a serviços restritos por usuários ilegítimos, objetivando ser ao mesmo tempo seguro e de fácil uso para humanos. O presente sistema permite que uma entidade atue simultaneamente como cliente e provedor de serviços, sem a necessidade de utilizar diretórios para anunciar serviços e/ou realizar buscas pelos mesmos, fazendo ainda com que a existência de provedores redundantes para um mesmo serviço seja tratada de forma a ser transparente para os usuários. Para validação, é avaliado o tráfego gerado pelas mensagens do Sec-SD e além disso, foi desenvolvido um protótipo, que é utilizado para integrar funcionalidades para descoberta segura de serviços no LP2P (*Local Peer-to-Peer Protocol*), que é uma plataforma para compartilhamento de arquivos P2P para redes locais desenvolvida no Grupo de Redes de Computadores e Sistemas Distribuídos do PIPCA – UNISINOS.

Palavras-chave: Sistemas distribuídos. Descoberta de serviços. Segurança. Criptografia. Autenticação.

ABSTRACT

Given the growing popularity of mobile devices in recent years, there is an increasing need for connectivity and services in computer networks. In this context, service discovery technologies aim to simplify the interaction between users and devices, facilitating administrative tasks, especially when there is a need to add new equipments. Due to the diversity of environments where these technologies can be used, there also the need to address security issues and, at the same time, to preserv the usability of the system. This work presents a system for secure service discovery on local networks with decentralized architecture, called Sec-SD – *Secure Service Discovery Protocol*. Sec-SD makes use of cryptography and authentication mechanisms in order to allow only valid users to obtain information about the available services. In this way, is possible to prevent several attacks caused by the advertisement of false services and by service access performed by illegitimate users, aiming to be at the same time secure and easy-of-use for humans. This model allows the same entity to act simultaneously as a client and a service provider, without the use of any directory to register services and/or search for available services. The existence of redundant service providers is also addressed by the model in order to be transparent for human users. For validation, the traffic generated by the Sec-SD messages is evaluated and a prototype is used to provide secure service discovery facilities into a P2P file sharing focused in Local Area Networks called LP2P (*Local Peer-to-Peer Protocol*), that was developed at PIPCA - UNISINOS.

Keywords: Distributed systems. Service discovery. Security. Cryptography. Authentication.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DNS–SD	Domain Name System Based Service Discovery
EAP	Extensible Authentication Protocol
FTP	File Transfer Protocol
GENA	General Event Notification Architecture
HTTP	Hypertext Transfer Protocol
HTTPU	Hypertext Transfer Protocol over UDP
IDEA	International Data Encryption Algorithm
IETF	Internet Engineering Task Force
IP	Internet Protocol
LP2P	Local Peer-to-Peer Protocol
mDNS	Multicast Domain Name System
MD5	Message Digest 5
NFS	Network File System
NIST	National Institute of Standards and Technology
OLSR	Optimized Link State Routing
RF	Rádio-Frequência
SHA-1	Secure Hash Algorithm 1
SOAP	Service Oriented Architectures and Programming
SSL	Secure Sockets Layer
SSDP	Simple Service Discovery Protocol
SLP	Service Location Protocol
TCP	Transmission Control Protocol
TLD	Top-Level Domain
TLS	Transport Layer Security
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
VRR	Virtual Ring Routing

XML Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

As tecnologias de computadores dos últimos anos têm sido marcadas por uma necessidade cada vez maior de conectividade em rede e pelo crescimento do uso de dispositivos móveis. Um exemplo dessas tendências é a recente popularização dos netbooks e dos smartphones, que combinam exatamente as duas características.

Diante desse crescimento, é natural que surja também uma maior necessidade de serviços nas redes de computadores. Serviços, nesse contexto, são recursos que podem ser utilizados, localmente ou à distância, por pessoas, programas, dispositivos ou mesmo por outros serviços (COULOURIS; DOLLIMORE; KINDBERG, 2005). Assim, podem ser definidas duas entidades: o provedor — que é quem disponibiliza o serviço — e o cliente, que é quem utiliza o serviço. Um dispositivo pode se comportar como provedor, cliente ou ambos, e ainda, oferecer diversos tipos de serviços. A Figura 1 mostra aplicações da descoberta de serviços – os dispositivos estão em rede na residência e no escritório e com o uso de um sistema para descoberta de serviços, os usuários podem ter conhecimento sobre os serviços oferecidos por cada um dos dispositivos.

Abaixo são relacionados alguns cenários onde a utilização de tecnologias para descoberta de serviços é desejável:

- Dispositivos conectados em uma residência inteligente;
- Um serviço de impressão em um escritório, que deve ser acessível por todas as máquinas;
- Compartilhamento de arquivos estratégicos entre executivos participando de uma reunião de negócios ou de uma conferência, conectados entre si com seus notebooks via rede wireless. Neste último caso, não seria prudente divulgar a existência dos arquivos a todas as máquinas da rede.

Conforme (EDWARDS, 2006), a descoberta de serviços pode ser caracterizada como um processo no qual um cliente em uma rede é notificado espontaneamente sobre a disponibilidade de recursos de seu interesse, podendo também procurar pelos mesmos sem necessitar da interferência de um administrador. Quando um recurso entra em uma rede, ele se torna disponível através do registro no sistema de descoberta. Nesta etapa são fornecidas informações que mais tarde possibilitarão que o serviço seja identificado como sendo de interesse para os clientes e então acessado. Desta forma, um cliente pode procurar por serviços fornecendo critérios para busca que são utilizados pelos provedores para verificar se os mesmos oferecem os recursos desejados pelo cliente. Assim, torna-se possível para os usuários encontrar e utilizar serviços em uma rede sem ter conhecimento prévio sobre a existência dos mesmos.

Do ponto de vista dos usuários, as aplicações para descoberta de serviços tornam mais fáceis as tarefas de encontrar e utilizar recursos em uma rede. Já do ponto de vista dos administradores, essas aplicações simplificam a construção e a manutenção de redes, especialmente no que diz

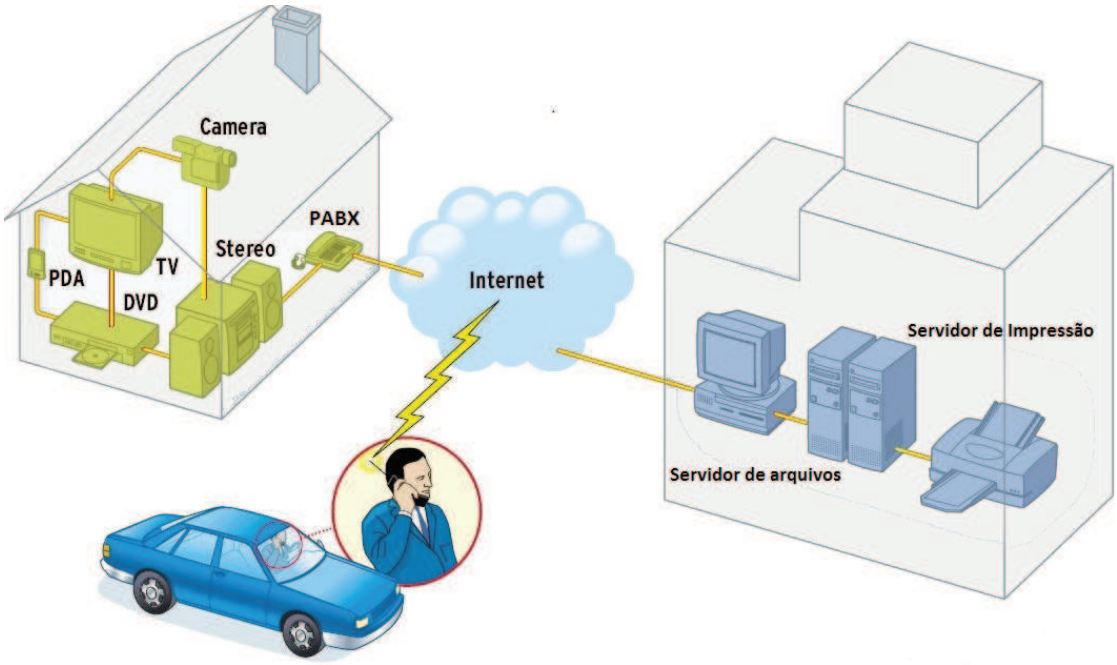

Figura 1: Aplicações da descoberta e uso de serviços

respeito à introdução de novos serviços e dispositivos. Estas tecnologias também são vantajosas do ponto de vista financeiro, uma vez minimizam a necessidade de atendimento por pessoal técnico, especialmente quando ocorrem mudanças (AUBINEAU, 2004).

1.1 Motivação

Como consequência natural do interesse nas aplicações para encontrar de forma automática os recursos disponíveis em redes de computadores, pode-se constatar atualmente uma grande diversidade de tecnologias e sistemas dedicados à publicação, descoberta e utilização de serviços. Estas tecnologias estão se tornando cada vez mais populares tanto para uso em redes de computadores convencionais, presentes em universidades e empresas, como uso em redes móveis, como o Wi-Fi (IEEE COMPUTER SOCIETY, 2007) e o Bluetooth (METTÄLÄ et al., 1999).

Nesse contexto, surge também a preocupação com as questões relacionadas a segurança. Em determinados cenários os requisitos de segurança são mínimos, como por exemplo a busca por impressoras em um escritório, onde os possíveis usuários são funcionários para os quais não são aplicadas restrições quanto ao uso do serviço. Já em outros cenários, como uma conferência, um usuário mal intencionado poderia anunciar uma falsa impressora com o objetivo de capturar documentos confidenciais.

Ainda, uma outra situação onde é necessário estabelecer uma relação de confiança entre usuário e provedor de serviços é o compartilhamento P2P de arquivos em ambientes acadêmicos e corporativos, onde é comum a existência de conteúdos de acesso restrito (GOLBECK, 2008).

Os exemplos anteriores caracterizam situações onde é exigido um nível alto de segurança para o processo de anúncio e descoberta de serviços, existindo claramente a necessidade de

comprovar a identidade dos usuários e dos provedores de serviços. Desta forma, a máquina cliente deve fornecer dados sobre sua identidade para que a máquina provedora possa divulgar informações relacionadas aos serviços disponibilizados sem comprometer a sua segurança e o cliente deve dispor de meios para comprovar que o provedor de serviços é quem afirma ser.

Diante da crescente preocupação com os aspectos de segurança nos sistemas de descoberta de serviços, surgiram nos últimos anos protocolos seguros para este fim e ainda, alternativas para integrar mecanismos de segurança em tecnologias já existentes, como o JINI[1] e o UPnP (PRESSER et al., 2008). As soluções existentes são resultados de esforços da indústria e do meio acadêmico, onde o interesse em tratar as questões de segurança na descoberta de serviços tem aumentado significativamente.

1.2 Definição do problema

Combinar mecanismos para atender os requisitos de segurança com tecnologias para descoberta e ao mesmo tempo, preservar a facilidade de uso do sistema é uma tarefa bastante complexa. De acordo com (ZHU et al., 2010), um sistema seguro deve impedir que um cliente ilegítimo tenha acesso a um determinado serviço e até mesmo que ele obtenha informações sobre os serviços oferecidos em certos casos, mas sempre deve permitir que um usuário que tenha as credenciais corretas tome conhecimento das características dos serviços e com isso, possa ter acesso aos mesmos.

Em determinadas situações, é necessário que apenas os usuários válidos saibam sobre a existência de determinados serviços, para prevenir ataques. Por exemplo, em um sistema de arquivos compartilhado utilizado em uma grande empresa, uma pasta que contenha arquivos relacionados a folha de pagamento deve ter o acesso restrito somente a diretoria e a alguns funcionários do setor administrativo/financeiro. Desta forma, seria mais prudente que outros usuários da rede não tivessem conhecimento sobre a localização e o conteúdo desta pasta. Por outro lado, os conteúdos de uso comum, como por exemplo pastas que contenham documentos com a política da empresa ou com roteiros de processos devem ser divulgados a todos os funcionários.

Além disso, é necessário que os serviços disponíveis na rede possam ser facilmente identificados pelos usuários, ou seja, a existência de recursos acessíveis a um usuário deve ser divulgada a ele através de uma interface amigável, onde estejam presentes todas as informações necessárias para que ele escolha os serviços que melhor atendem as suas necessidades. Esta facilidade de uso deve ser preservada mesmo com a utilização de mecanismos de segurança, onde descoberta ocorra de forma transparente para o usuário humano, com uma quantidade mínima de ações do mesmo. Por exemplo, não é desejável que um usuário necessite entrar com dados relacionados a sua identidade a cada sessão de descoberta de serviços para que o processo possa ser realizado com segurança.

[1] http://www.jini.org

Outra questão relevante em relação ao projeto de protocolos de descoberta é a quantidade de mensagens trocadas durante a negociação entre usuários e provedores de serviços. Esta característica deve ser levada em conta também no desenvolvimento de protocolos inseguros, mas torna-se ainda mais crítica quando são utilizados mecanismos de segurança, que envolvem de modo geral procedimentos como o envio de senhas, certificados, troca de chaves, etc.

Ainda que a descoberta de serviços possa ser realizada de forma distribuída, as funcionalidades relacionadas a segurança desse processo são muitas vezes desempenhadas por servidores centrais (ex: tecnologia JINI[2]). Dessa forma, o uso de descentralização para funções como a autenticação de entidades na descoberta de serviços também representa uma questão de pesquisa.

1.3 Objetivos

O objetivo deste trabalho é o desenvolvimento de um sistema seguro para descoberta de serviços em redes locais com arquitetura distribuída — o Sec-SD (*Secure Service Discovery Protocol*).

A motivação para o mesmo surgiu da necessidade de combinar requisitos de segurança em um protocolo para descoberta de serviços voltado aos ambientes locais e com arquitetura distribuída, sem o uso de diretórios, onde todos os dispositivos possam atuar como cliente e provedor de serviços simultâneamente, podendo existir provedores redundantes. Ao mesmo tempo, busca-se preservar a facilidade de uso do sistema, fazendo com que o processo de descoberta ocorra de forma transparente para usuário humanos.

Como caso de uso, o Sec-SD é executado em conjunto com o sistema LP2P (*Local Peer-to-Peer*) (ROCHA; MARCON; ÁVILA, 2010) em desenvolvimento no grupo de Redes de Computadores e Sistemas Distribuídos do PIPCA. O LP2P propõe um sistema de compartilhamento peer-to-peer para uso em redes locais, e seu projeto leva em consideração as características específicas deste tipo de rede, como a alta taxa de transmissão e a baixa latência de comunicação.

Do ponto de vista do LP2P, serviços são compartilhamentos, que são pastas com conteúdo distribuído entre os nodos da rede, e podem ser de livre acesso aos usuários ou restritos. A descoberta de compartilhamentos restritos deve ser limitada aos usuários que possuem as credenciais necessárias, ao mesmo tempo em que os conteúdos abertos devem ser divulgados para todos os usuários no momento em que se tornam disponíveis na rede.

São objetivos específicos deste trabalho:

- O desenvolvimento e a implementação de um protocolo para descoberta segura de serviços com arquitetura distribuída e voltado aos ambientes locais.

- O protocolo deve atuar sem o uso de diretórios para registro de serviços e buscas pelos mesmos. Assim, cada entidade deve ser responsável por gerenciar os serviços disponibi-

[2] http://www.jini.org

lizados, as credenciais necessárias para buscar por serviços oferecidos por outros provedores e os mecanismos necessários para autenticação.

- Deve ser suportada a existência de provedores de serviço redundantes, que é empregada com o intuito de assegurar a disponibilidade de determinados serviços. Neste caso, o protocolo deve tratar a redundância de servidores de modo a torná-la transparente para o usuário humano, que na presença de um ou mais provedores para um dado serviço deve visualizar um único serviço na rede.

- O protocolo também deve atuar de forma a evitar a exposição de informações confidenciais na descoberta de serviços restritos. Para isso, os provedores devem aguardar o recebimento de requisições e na presença delas, verificar se o cliente possui credenciais válidas para acessar o serviço. Nesse caso os provedores não devem, em hipótese alguma, enviar anúncios espontâneos.

- A procura por serviços restritos deve ocorrer sem que o usuário envie requisições em claro, para evitar ataques e exposição de informações confidenciais;

- Como resultado de sua atuação, o protocolo deve estabelecer uma relação de confiança entre usuário e o provedor na descoberta de serviços confidenciais, fazendo com que ambos tenham certeza de que o outro realmente é quem ele afirma ser;

Diante da grande quantidade de aspectos a serem tratados no projeto de tecnologias para descoberta de serviços, originados principalmente pelas diferenças entre os ambientes de funcionamento, tiveram de ser impostas algumas limitações para este trabalho. Assim, os seguintes aspectos não são contemplados:

- O Sec-SD não provê mecanismos para acesso aos serviços, mas garante que apenas os clientes que possuem as devidas credenciais poderão localizar e utilizar os serviços. Por exemplo, no sistema LP2P, o acesso aos serviços (como por exemplo, a cópia de um arquivo) é feito através do próprio protocolo LP2P;

- As credenciais para acesso as informações dos serviços devem ser repassadas somente aos interessados e através de um meio seguro. Entretanto, a escolha desse meio é de responsabilidade exclusiva dos usuários e está fora do escopo desse trabalho;

- O Sec-SD tem como escopo de funcionamento a rede local. Desta forma, seus requisitos visam atender bem as necessidades das aplicações projetadas para este fim, como é o caso do sistema LP2P.

O uso de mecanismos de segurança em aplicações como o LP2P torna-se necessário devido às características deste ambiente. Por exemplo, a entrada e saída de usuários (e consequentemente de conteúdos) ocorre com certa frequência, fazendo com que exista preocupação com a

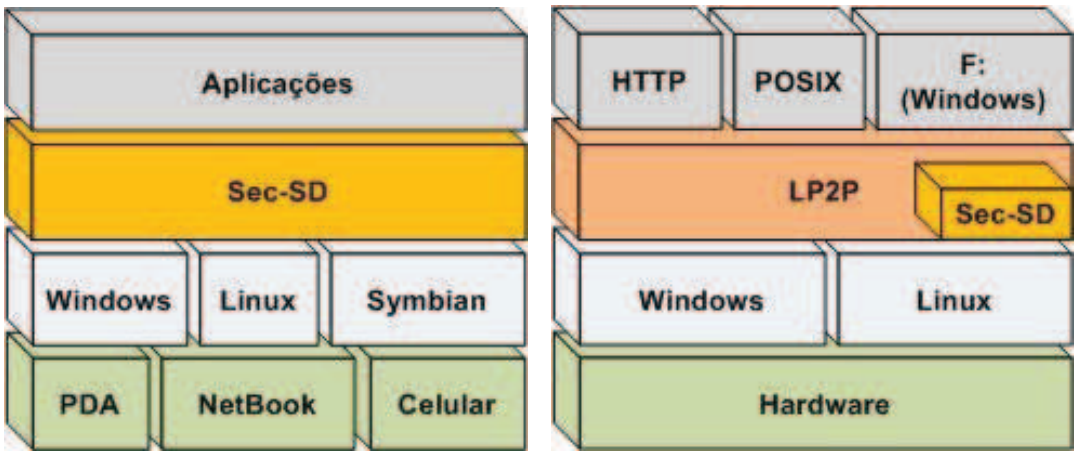

Figura 2: Sec-SD em um ambiente de dispositivos móveis

privacidade, uma vez que não são todos os arquivos que podem ser compartilhados com todos os usuários. Assim, deve ser possível garantir que os conteúdos confidenciais serão disponibilizados somente para os usuários que possuem credenciais válidas. Por outro lado, um cliente que procura por um arquivo de caráter confidencial deve ter a certeza de que o provedor é mesmo quem afirma ser.

A Figura 2 ilustra o enquadramento do Sec-SD em diferentes cenários de aplicação — em dispositivos móveis e no LP2P — no segundo caso, atuando como uma camada intermediária entre os sistemas operacionais e as interfaces de acesso previstas (pastas do Windows, acesso via HTTP, entre outros).

Um cenário de uso para o Sec-SD é ilustrado na Figura 3. Neste caso, as máquinas PC1 e PC2 não disponibilizam serviços, mas são clientes dos serviços Página Web e Arquivos disponibilizados por outro provedor, o Servidor de arquivos e Web, que também atua como cliente da Impressora. PC1 e PC2 também tem conhecimento do serviço Fotos_Férias, que é provido pelo Tablet, que também é cliente da Impressora, que não tem acesso a serviços de outros provedores. As máquinas da rede visualizam listas de serviços diferentes, pois um cliente tem conhecimento somente dos serviços para os quais ele possui credenciais.

1.4 Organização do texto

O restante do texto está organizado da seguinte forma:

- No capítulo 2 estão os conceitos necessários para o entendimento deste trabalho, como por exemplo as arquiteturas empregadas nos sistemas de descoberta de serviços e os ataques à segurança mais comuns, bem como os requisitos de segurança e os mecanismos utilizados para evitá-los;

- O capítulo 3 apresenta tecnologias utilizadas para descoberta de serviços em redes com e sem infra-estrutura;

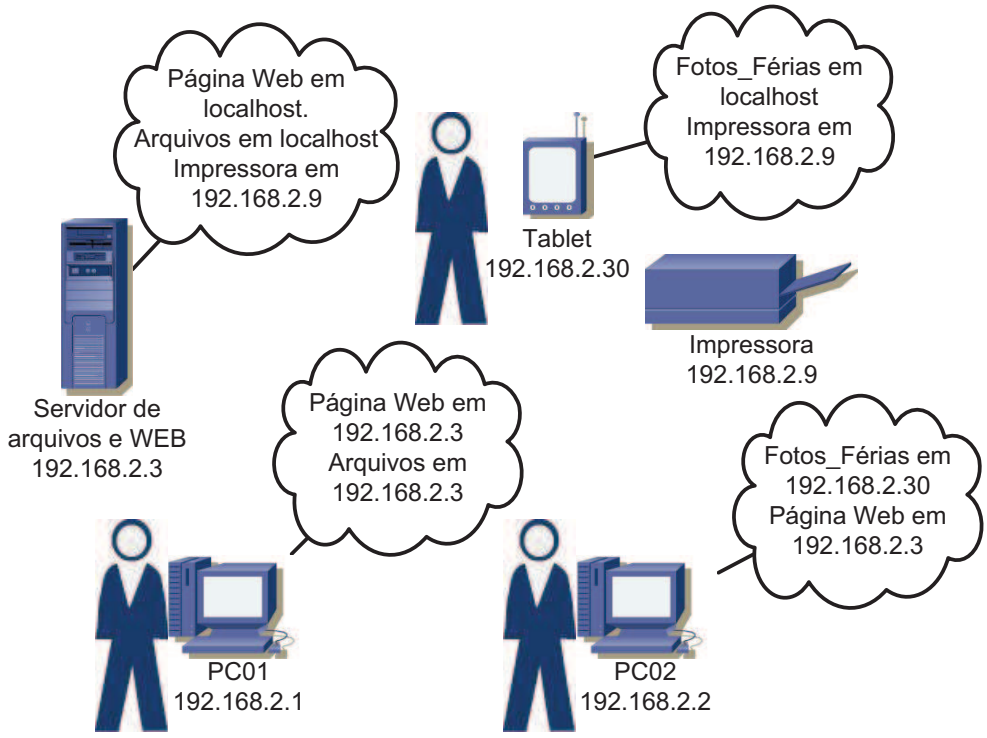

Figura 3: Cenário de uso para o Sec-SD.

- O capítulo 4 descreve as entidades do protocolo para descoberta de serviços, o funcionamento do Sec-SD e o protocolo em sí;
- O capítulo 5 apresenta a validação do Sec-SD, feita através da análise do tráfego gerado pelas mensagens de descoberta e da implementação do sistema e do uso de um protótipo. Também são apresentadas a metodologia e as tecnologias utilizadas para desenvolvimento do Sec-SD. Por fim, este capítulo apresenta uma visão geral do protocolo LP2P, escolhido como caso de uso para o Sec-SD e os resultados obtidos com este experimento;
- No capítulo 6 são apresentadas as conclusões e os trabalhos futuros.

2 CONCEITOS BÁSICOS

Este capítulo apresenta conceitos relacionados à descoberta de serviços, como por exemplo as diferentes arquiteturas que são empregadas, bem como conceitos relacionados à segurança, tais como os tipos de ataques, os requisitos necessários para evitá-los e os principais mecanismos de segurança utilizados na atualidade, como criptografia e assinaturas digitais. O texto deste capítulo é de grande importância para o entendimento deste trabalho.

2.1 Arquiteturas empregadas em sistemas de descoberta de serviços

De acordo com (EDWARDS, 2006), a descoberta de serviços é um processo no qual um cliente em uma rede é notificado espontaneamente sobre a disponibilidade de recursos e pode procurar pelos mesmos sem a interferência de um administrador. No momento em que um recurso ingressa em uma rede, ele se registra no sistema de descoberta para permitir que os clientes saibam que ele está disponível. Ele deve fornecer informações para facilitar o seu reconhecimento pelos clientes. Por exemplo, uma impressora pode informar ao sistema de descoberta que imprime colorido e que se encontra no setor financeiro da empresa XYZ. Desta forma, um cliente pode procurar por serviços fornecendo critérios para busca e os provedores fazem uso destes critérios para verificar se eles oferecem ou não o serviço desejado pelo cliente. Deste modo, torna-se possível que os usuários encontrem serviços em uma rede sem ter conhecimento prévio sobre a existência dos mesmos.

Os sistemas de descoberta de serviços podem ser implementados de forma centralizada, distribuída ou híbrida (VERVERIDIS; POLYZOS, 2008) (ZHU; MUTKA, 2005), conforme é explicado a seguir.

Arquitetura centralizada Neste caso existem diretórios responsáveis pela realização de anúncios e consultas em nome de todos os dispositivos da rede. Os serviços são registrados pelos seus provedores junto ao diretório.

Arquitetura distribuída Cada dispositivo é responsável por anunciar na rede e manter atualizadas as informações relacionadas aos serviços que ele disponibiliza ou conhece.

Arquitetura híbrida Faz uso de diretórios para divulgação de serviços e busca pelos mesmos quando existem diretórios disponíveis e permite a interação direta entre clientes e provedores quando não há diretórios.

A Figura 4 ilustra o processo de descoberta para a descoberta centralizada e distribuída. No modo centralizado, o provedor de serviços informa ao diretórios sobre os seus serviços e as características dos mesmos (como por exemplo, Wiki com as informações do projeto X, da equipe A, no IP a.b.c.d). Enquanto isso, um cliente pode pedir ao diretório que o informe sobre todos os servidores Web disponíveis na sua rede. Já no modo distribuído, o provedor anuncia

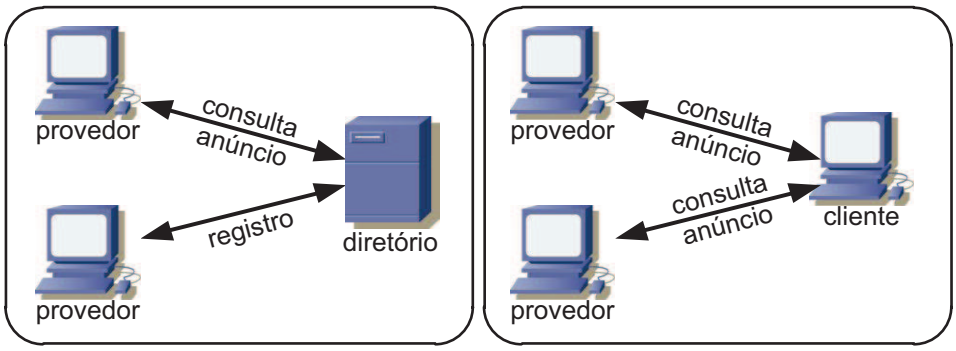

Figura 4: Registro no sistema de descoberta com arquitetura centralizada e distribuída

aos clientes os serviços que ele disponibiliza assim que ingressa na rede (por exemplo, usando mensagens multicast). Do mesmo modo, os clientes podem enviar requisições diretamente para os provedores.

No que diz respeito as maneiras de registrar serviços e obter informações sobre os serviços disponíveis, existem três formas básicas de operação (VERVERIDIS; POLYZOS, 2008).

Reativo ou baseado em requisição Neste modo, os usuários enviam requisições aos diretórios ou diretamente aos provedores, dependendo da arquitetura empregada e as solicitações são respondidas imediatamente e sob demanda.

Pró-ativo ou baseado em anúncio Os provedores de serviço anunciam periodicamente os seus serviços aos diretórios ou diretamente aos potenciais clientes.

Híbrido Os modos reativo e pró-ativo operam de forma combinada. Por exemplo, os provedores podem anunciar os seus serviços aos diretórios (modo pró-ativo) e os clientes podem encaminhar suas requisições aos diretórios (modo reativo).

As arquiteturas utilizadas nos sistemas de descoberta de serviços, bem como os modos de registro de serviços e busca pelos mesmos que são utilizados nos sistemas atualmente são explicados no capítulo 3.

2.2 Segurança

Esta seção apresenta conceitos relacionados à segurança, tais como os tipos de ataques, os requisitos necessários para evitá-los e os mecanismos de segurança utilizados atualmente. Estes conceitos aplicam-se bem aos sistemas de descoberta de serviços mas não se restringem a eles.

2.2.1 Ataques à segurança

Ataques à segurança são ações realizadas com o objetivo de capturar informações pertencentes a uma organização ou ainda, para prejudicar ou impossibilitar o funcionamento de

um sistema. No primeiro caso, trata-se de um ataque passivo e no segundo, de um ataque ativo (TANENBAUM, 2003) (STALLINGS, 2008).

Os ataques passivos não alteram dados e por este motivo raramente são detectados (como por exemplo, análise de tráfego em uma rede), devendo portanto ser prevenidos. Já os ataques ativos causam modificação nos dados e/ou envio de fluxos falsos, que podem ser obtidos através da repetição de informações previamente capturadas. Um exemplo de ataque ativo é o ataque do ”homem do meio” (ou *man-in-the-middle attack*) onde uma entidade intercepta mensagens, modifica e repassa ao verdadeiro destinatário. Também é possível que uma entidade finja ser um usuário autorizado a fazer uso de recursos de um determinado sistema. Um outro tipo de ataque ativo é o de negação de serviço (ou *Denial of Service, DoS*), onde uma entidade impede o funcionamento de um sistema ou de parte dele. Um exemplo disso é a abertura de muitas conexões com o servidor de um site de comércio eletrônico por parte de um concorrente para deixar o sistema inacessível.

2.2.2 Requisitos de segurança

Os requisitos de segurança necessários para impedir ataques passivos e ativos (STALLINGS, 2008) (FOROUZAN, 2008) são listados abaixo:

- **Integridade** Um fluxo de informações com garantia de integridade deve ser entregue ao destinatário exatamente como foi enviado, ou seja, sem que ocorra qualquer alteração, acidental ou não, durante a transmissão dos dados. Uma modificação no conteúdo da mensagem é detectada facilmente pelo receptor. Exemplos de aplicações: Em uma transferência bancária é necessário garantir que o valor informado pelo cliente foi recebido corretamente pelo banco antes que este confirme a operação. A verificação da integridade de uma mensagem também protege contra ataques do tipo ”homem do meio”, pois se uma terceira parte alterar uma mensagem e repassá-la ao destinatário original, este irá perceber facilmente a alteração feita.

- **Confidencialidade** Uma mensagem de caráter confidencial deve ser entendida apenas pelo emissor e pelo receptor. Se o tráfego deste tipo for interceptado, a informação apresentada ao intruso deve ser incompreensível. Exemplo de aplicação: Uma senha enviada a um servidor para acessar uma conta de e-mail não deve ser compreendida se for capturada.

- **Autenticação de mensagens** Permite que o receptor verifique a origem de uma mensagem, a fim de garantir que a informação realmente foi enviada pela entidade que afirma ser o remetente e não por um impostor.

- **Autorização** Utilizada para garantir que apenas os usuários que possuem as devidas permissões terão acesso aos recursos de um sistema. Por exemplo, um usuário pode ter

autorização para ler as configurações de um equipamento de rede sem modificá-las.

- **Irretratabilidade** Garante que um remetente não poderá negar o envio de uma mensagem, assim como o destinatário não poderá afirmar que não recebeu tal informação.

- **Disponibilidade** Está relacionada ao fato de que um determinado recurso deve estar disponível para as entidades que possuem autorização para utilizá-lo, ou seja, um serviço de disponibilidade deve proteger o sistema no qual ele atua contra ataques de negação de serviço.

2.2.3 Mecanismos de segurança

Os requisitos de segurança descritos anteriormente podem ser providos com a utilização de determinados mecanismos que são descritos a seguir.

2.2.3.1 Criptografia simétrica e assimétrica

A criptografia (STALLINGS, 2008) é conjunto de princípios e técnicas utilizados para proteger informações confidenciais. É utilizada desde a antiguidade para ocultar mensagens e um dos primeiros métodos empregados para este fim, conhecido como Cifra de César, consiste na substituição de cada letra do alfabeto pela letra que está três posições a frente. Atualmente, os algoritmos utilizados possuem forte embasamento matemático e podem ser classificados em duas categorias: os algoritmos de criptografia simétrica (ou de chave secreta) e os algoritmos de criptografia assimétrica (ou de chave pública).

Na criptografia de chave simétrica é utilizada uma mesma chave para cifrar o texto no lado do emissor e para decifrá-lo no lado do receptor. Já na criptografia assimétrica, o remetente utiliza a chave pública do destinatário para cifrar a mensagem e o receptor, de posse da sua chave privada, pode decifrar a informação recebida. É importante destacar que no primeiro caso, a chave deve ser conhecida apenas pelas partes envolvidas na comunicação. Do contrário, o sigilo dos dados trocados entre as entidades estará comprometido. Já na criptografia assimétrica, a chave pública (como o próprio nome diz) é de domínio público, enquanto que a chave privada deve ser armazenada de modo seguro pelo seu proprietário.

São exemplos de algoritmos de criptografia simétrica o o AES (*Advanced Encryption Standard*) (NIST, 1997), o DES (*Data Encryption Standard*) (NIST, 1999), o IDEA (*International Data Encryption Algorithm*) (SCHNEIER, 1995) e o Blowfish (SCHNEIER, 1993) e para criptografia assimétrica, RSA (*Rivest, Shamir, Adleman*) (JONSSON; KALISKI, 2003) e Diffie-Hellman (RESCORLA, 1999).

2.2.3.2 Hash

Uma função de hash gera um resumo da mensagem, que é um padrão de bits calculado de modo que torna muito difícil encontrar duas mensagens com o mesmo hash. Além disso, deve ser impossível obter a informação original a partir do resumo gerado (STALLINGS, 2008).

Os algoritmos de hash SHA-1 (*Secure Hash Algoritm 1*) e MD5 (*Message-Digest algorithm 5*) são bastante utilizados, tendo sido adotados, por exemplo, no padrão X.509 de certificados digitais (HOUSLEY et al., 1999). O SHA-1 foi lançado em 1995 a partir de uma revisão do algoritmo SHA, desenvolvido pelo NIST (*National Institute of Standars and Technology*) dos EUA e publicado originalmente em 1993. Este algoritmo produz valores de hash com 160, 256, 384 e 512 bits. Para todos os tamanhos o processo utilizado é basicamente o mesmo. É criado um hash de comprimento N a partir de uma mensagem com diversos blocos, cada um com comprimento igual a 512 bits (para SHA-160 e 256) ou 1024 bits (para SHA-512 e 1024). O MD5 utiliza quatro rodadas. Em cada rodada é aplicada uma entre quatro funções não-lineares em cada um dos segmentos de 32 bits contidos em um bloco de 512 bits de texto, resultando em um hash de 128 bits. No que diz respeito ao desempenho, o MD5 é significativamente mais rápido do que SHA-1.

2.2.3.3 Assinatura digital

A assinatura digital permite que o receptor verifique a procedência de uma mensagem recebida, assim como acontece com uma assinatura convencional em um documento, que tem como função provar que o mesmo é autêntico. Ao receber uma mensagem assinada digitalmente, o destinatário verifica se a assinatura é realmente da entidade que afirma ter enviado tal informação e se isso não for provado, ele descarta a mensagem. Diferentemente do que acontece com uma assinatura convencional, onde uma pessoa utiliza a mesma assinatura para diversos documentos, cada mensagem deve ter sua própria assinatura digital (TANENBAUM, 2003).

O remetente utiliza sua chave privada para assinar a mensagem, aplicando esta chave a um algoritmo de assinatura, enquanto que o destinatário verifica a procedência da informação recebida com o uso da chave pública do remetente aplicada a um algoritmo de verificação. Existe a possibilidade de o signatário assinar a mensagem inteira ou apenas um resumo (ou hash) da mesma, que é o método mais utilizado atualmente. Em ambos os casos, o uso de assinatura digital também provê mecanismos para verificação da integridade da mensagem, pelo fato de que é muito difícil obter a mesma assinatura caso a mensagem seja alterada. É importante ressaltar que o uso de assinatura digital não provê confidencialidade, uma vez que é utilizada a chave pública da entidade que envia a mensagem para comprovar a autenticidade e, desta forma, o conteúdo transmitido pode ser verificado por qualquer um.

Para assinar a mensagem inteira, o emissor criptografa a mesma com a sua chave privada e o receptor faz a verificação descriptografando a informação recebida com a chave pública

do suposto emissor. Para assinar o hash da mensagem, o mesmo é criptografado com a chave privada do remetente. Do outro lado, o receptor aplica a mesma função de hash na mensagem recebida e descriptografa a assinatura com uso da chave pública do emissor, comparando os dois resultados. Se forem iguais, a mensagem é autêntica e deve ser aceita, do contrário, deve ser descartada.

2.2.3.4 Exemplos de protocolos seguros utilizados comercialmente

A seguir são mostrados exemplos de aplicações que fazem uso dos mecanismos de segurança explicados anteriormente. Os sistemas de descoberta de serviços que fazem uso de segurança são tratados no capítulo seguinte.

HTTPS

O HTTPS *(HyperText Transfer Protocol Secure)* (RESCORLA, 2000) é a implementação do protocolo HTTP (FIELDING et al., 1999) sobre TLS *(Transport Layer Security)* (DIERKS; RESCORLA, 2008) e tem como objetivo de permitir que os dados sejam transmitidos por meio de uma conexão criptografada. Além disso, seu uso permite que seja verificada a autenticidade do cliente e do servidor por meio do uso de certificados digitais.

No HTTPS, a entidade que atua como cliente HTTP também deve atuar como o cliente TLS, necessitando iniciar uma conexão com o servidor na porta apropriada e em seguida iniciar o *handshake* TLS. Ao final deste, o cliente pode iniciar a primeira solicitação HTTP. Os dados HTTP devem ser enviados como dados de aplicação TLS e o comportamento normal do HTTP deve ser seguido.

O HTTPS é comumente utilizado em serviços de webmail, comércio online e sites de bancos, entre outros.

RADIUS

O RADIUS *Remote Authentication Dial-In User Service* (RIGNEY et al., 2000) (RIGNEY, 2000) é utilizado para prover serviços de autenticação, autorização e estatísticas *(accounting)* e faz uso de UDP *(User Datagram Protocol)* na comunicação entre cliente e servidor.

Um cliente RADIUS (que pode ser encontrado por exemplo em pontos de acesso para redes sem fio, switches e roteadores) envia informações sobre as credenciais dos usuários para o servidor, sendo que a senha do usuário é enviada criptografada com um segredo que é compartilhado entre cliente e servidor. É realizada então a verificação a solicitação do cliente por parte do servidor, que retorna uma resposta que pode ser de rejeição ou de aceitação, podendo conter também informações de autorização. A interação entre estas entidades é mostrada na Figura 5.

Os clientes podem também enviar mensagens sobre estatísticas para o servidor com o obje-

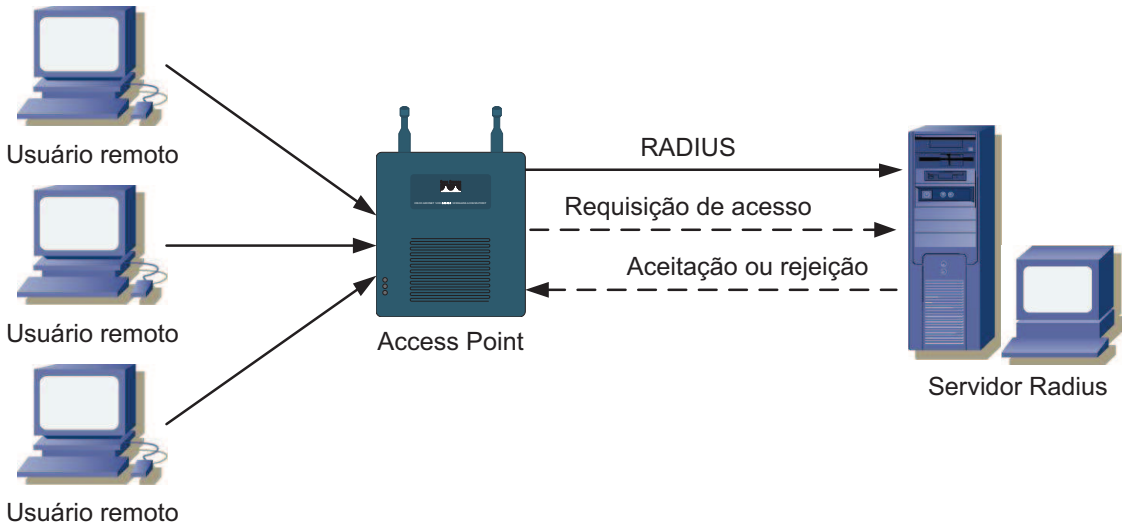

Figura 5: Comunicação entre cliente e servidor RADIUS.

tivo de informar as suas ações, como por exemplo, a execução de comandos em um equipamento da rede enviando também o momento da ocorrência.

Como exemplos de servidores RADIUS podem ser citados o FreeRADIUS[1] e o Cisco ACS *(Access control System)*[2].

TACACS+

O TACACS+ *(Terminal Access Controller Access Control System)* (D. CARREL, 1997) fornece serviços de autenticação, autorização e estatísticas e, diferentemente do RADIUS, utiliza TCP na comunicação entre cliente e servidor, que também compartilham um segredo. No TACACS+, o segredo é usado para criptografar todo o corpo das mensagens.

O cliente TACACS+ envia para o servidor uma requisição de autenticação (onde é enviado primeiramente o nome de usuário) e o servidor envia um pedido de senha. O cliente envia então a senha do usuário e recebe a resposta do servidor (autenticado ou não autenticado).

Ao contrário do que acontece com o RADIUS, a resposta não contém informações de autorização. Para cada ação que o usuário desejar executar deve ser enviada uma requisição de autorização pelo cliente. Nas requisições de armazenamento de estatísticas *(accounting)*, consta a ação executada e o momento exato em que esta ocorreu.

São exemplos de servidores TACACS+ o Tac Plus[3] e o Cisco ACS.

[1] http://www.freeradius.org
[2] http://www.cisco.com/en/US/products/sw/secursw/ps5338/index.html
[3] http://www.gazi.edu.tr/tacacs

3 TRABALHOS RELACIONADOS

Existe uma grande quantidade de trabalhos realizados com o objetivo de viabilizar a descoberta de serviços em redes de computadores e/ou dispositivos móveis. Neste capítulo são apresentados os resultados de um estudo das tecnologias desenvolvidas para este fim resultantes de esforços da indústria e do meio acadêmico e que são mais relevantes do ponto de vista deste trabalho.

O estudo de tais tecnologias foi realizado com o objetivo de identificar os mecanismos de segurança empregados atualmente nos protocolos para descoberta de serviços, as questões referentes a segurança que não são tratadas por tais protocolos, a escalabilidade dos mesmos, a arquitetura empregada e a facilidade de uso, ou seja, o quão transparente é o processo de descoberta para o usuário humano. Ao final deste capítulo são comentadas as características das tecnologias estudadas que são mais importantes do ponto de vista deste trabalho.

3.1 SLP – Service Location Protocol

O SLP (IETF, 1999) foi definido pelo IETF (*Internet Engineering Task Force*) e projetado para redes IP. Sua primeira versão foi publicada em 1997 e a segunda em 1999. Esta tecnologia trabalha com a noção de escopo, que pode ser, por exemplo, um departamento dentro de uma companhia. O modo como a descoberta de serviços é realizada pode ser centralizado, onde existem diretórios responsáveis pela realização de anúncios e consultas em nome de todos os dispositivos da rede, ou distribuída, onde cada dispositivo é responsável por anunciar na rede e manter atualizadas as informações relacionadas aos serviços que ele disponibiliza ou conhece.

Na arquitetura distribuída existem duas entidades principais: os *User Agents — UAs* e os *Service Agents — SAs*. Os UAs, que são clientes dos serviços, enviam requisições aos SAs através de multicast e os SAs enviam respostas por meio de unicast. Os SAs podem anunciar seus serviços aos UAs por multicast. Já na abordagem centralizada existe uma entidade adicional: o *Directory Agent — DA*, que implementa um repositório de serviços. O endereço IP do DA pode ser conseguido de três diferentes formas: através de DHCP – neste caso o servidor de DHCP fornece o endereço quando recebe requisições; através dos anúncios feitos periodicamente pelos próprios DAs ou ainda, através de requisições feitas pelos UAs e SAs para o grupo multicast do SLP — um DA que esteja escutando irá receber a requisição e enviar uma resposta em unicast ao agente requisitante.

As Figuras 6 e 7 mostram o funcionamento básico do SLP com arquitetura centralizada e distribuída, respectivamente.

Os UAs enviam consultas ao DA, que responde com uma listagem (que pode ser vazia) contendo todas as URLs de SAs que correspondem à requisição. Esta lista é feita com base nas informações armazenadas, que são obtidas quando os SAs anunciam seus serviços. Os SAs precisam renovar de tempos em tempos as informações armazenadas pelos UAs ou DAs.

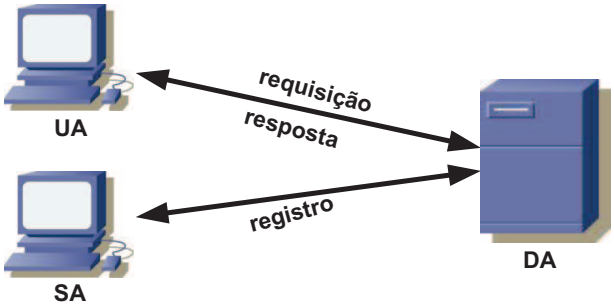

Figura 6: SLP com arquitetura centralizada

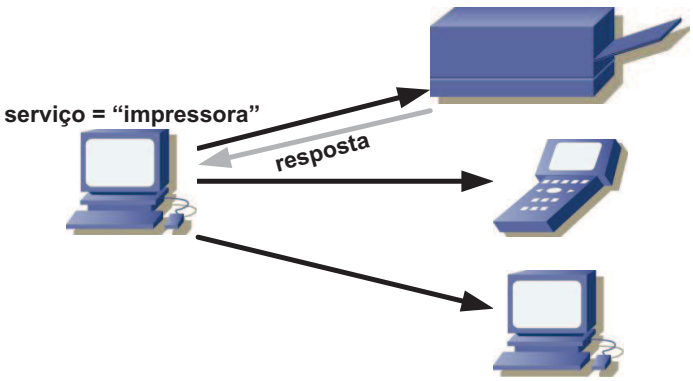

Figura 7: SLP com arquitetura distribuída

Pode existir mais de um DA no mesmo sistema. A comunicação entre eles e os outros agentes (UAs e SAs) pode ser via multicast ou unicast. Os DAs e SAs podem também se anunciar periodicamente na rede. Os SAs anunciam serviços através de URLs, que contêm informações como o tipo e os atributos do serviço em questão. Os UAs usam essas URLs para se comunicar diretamente com o serviço.

No que diz respeito a segurança, o SLP v2 provê um mecanismo de autenticação de uso opcional, cujo funcionamento é baseado em criptografia assimétrica e tem como objetivo permitir o estabelecimento de relações de confiança entre os SAs e os DAs e entre os UAs e DAs. Os SAs podem incluir assinaturas digitais nas suas mensagens de registro, sendo que o DAs verificam os dados recebidos antes de registrar ou excluir informações sobre serviços. As assinaturas são repassadas aos UAs, que podem rejeitar mensagens sem assinatura ou com assinatura incorreta. O SLP mostrou-se inadequado para redes com muitos nodos, devido ao tráfego gerado e, por esse motivo, seu projeto foi abandonado em 1999.

3.2 Jini

Desenvolvido pela Sun Microsystems, o Jini[1] é uma extensão da linguagem de programação Java e define regras sobre como dispositivos podem se conectar a outros dispositivos de modo a formar uma rede ad hoc simples chamada de comunidade Jini e ainda, possibilitar que seus serviços sejam utilizados por outros dispositivos da rede.

O Jini define três entidades: Provedores de serviço (*Service Providers - SP*), Serviço de Consulta (*Lookup Service - LS*) e Clientes. Além disso, existe o conceito de grupo, que é similar ao escopo do SLP. Para se registrar em uma rede Jini, um dispositivo precisa se inserir em uma base de dados (*Lookup Table*) de um LS. Esta base de dados pode conter além de ponteiros, interfaces Java para os serviços, que incluem os métodos que usuários e aplicações irão invocar para poder executá-lo, juntamente com outros atributos descritivos. O provedor do serviço localiza um LS na rede local através de requisições multicast e, então, se registra junto a ele.

Um cliente requisita um serviço de um tipo especifico e, para interagir com o mesmo deverá fazer uma cópia do código-objeto armazenado na *Lookup Table*. Isto elimina a necessidade de instalar previamente drivers para o serviço. O Jini utiliza três diferentes protocolos de descoberta, de acordo com cada caso. O protocolo de requisição multicast é utilizado quando uma aplicação ou serviço precisa procurar por um LS. Já o protocolo de anúncio multicast é usado quando um LS se anuncia aos serviços. Por fim, o protocolo de descoberta unicast é útil para estabelecer comunicações com um LS específico. As Figuras 8, 9 e 10 mostram respectivamente os processos de registro, descoberta e uso de um serviço.

Em uma rede Jini o usuário requisita um serviço por um período de tempo e, após fazer negociações com o provedor, tem o acesso ao serviço garantido pelo intervalo requisitado. Esta

[1] http://www.jini.org

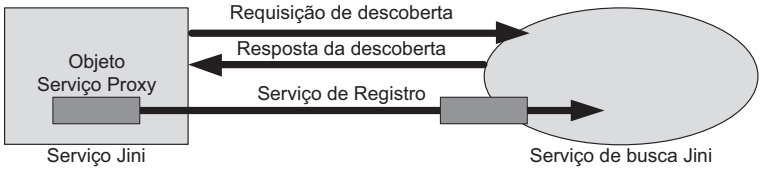

Figura 8: Registro de um serviço em Jini

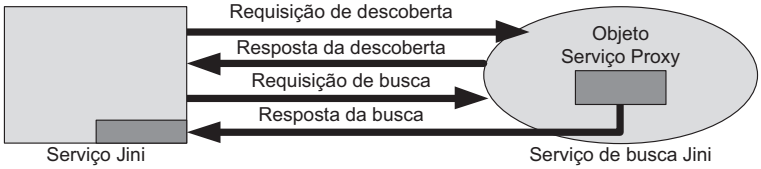

Figura 9: Descoberta de um serviço em Jini

licença pode ser renovada antes de expirar. Existe ainda a possibilidade de um dispositivo deixar a rede ou falhar de modo abrupto. Neste caso, quando a licença expira o dispositivo é excluído dos serviços de busca. Este modo de operação permite que o sistema funcione mesmo se ocorrer travamento de um serviço de consulta.

A segunda versão do Jini provê mecanismos de segurança, como criptografia, autenticação e integridade.

As requisições unicast para descoberta contém listagens com os identificadores dos formatos de descoberta propostos pelo cliente. Quando um Lookup Service que suporta a versão 2 do Jini recebe uma requisição, ele verifica se possui suporte a algum destes formatos e envia uma resposta que pode conter um identificador nulo se a comunicação não pode ser estabelecida de acordo com o que foi proposto pelo cliente.

Os possíveis formatos de mensagens incluem texto em aberto e mensagens autenticadas para as requisições e anúncios multicast. Para a descoberta unicast podem ser utilizados texto em aberto, TLS/SSL (DIERKS; RESCORLA, 2008) e Kerberos versão 5 GSS-API (LINN, 1996) e , sendo que para estes dois últimos os formatos das mensagens são bastante semelhantes. O Kerberos provê mecanismos para criptografia, autenticação e verificação da integridade dos dados.

Figura 10: Uso de um serviço em Jini

3.3 UPnP

O UPnP (PRESSER et al., 2008) foi desenvolvido por um consórcio de empresas fundado pela Microsoft e pensado para possibilitar a auto configuração de dispositivos e descoberta de serviços em pequenas redes, como de residências e escritórios.

A arquitetura UPnP define dois tipos de dispositivos: os dispositivos controlados e os pontos de controle. Um dispositivo controlado funciona como um um servidor, respondendo as requisições dos pontos de controle. Estes componentes podem ser implementados em uma grande variedade de plataformas, incluindo PCs e sistemas embarcados.

O UPnP utiliza o SSDP (*Simple Service Discovery Protocol*) (GOLAND et al., 1999) para anunciar a presença de um dispositivo para os outros participantes da rede e para descobrir os serviços que estão disponíveis. O SSDP usa HTTP multicast sobre UDP (HTTPU). Quando um dispositivo ou ponto de controle se conecta a rede, ele procura por um servidor DHCP. Na presença do servidor, o componente utiliza o endereço IP atribuído a ele por DHCP. Do contrário, ele escolhe um IP de uma faixa especial, podendo se mover entre redes com ou sem infraestrutura.

Os pontos de controle e dispositivos controlados podem ainda ter mais de uma interface de rede e mais de um endereço IP atribuído a cada interface. Quando um dispositivo controlado entra na rede, ele anuncia os seus serviços aos pontos de controle através do protocolo de descoberta UPnP. De modo semelhante, quando um ponto de controle é conectado a rede, ele procura por dispositivos de seu interesse. O processo de descoberta é mostrado na Figura 11.

Quando um dispositivo entra na rede, ele envia uma mensagem de anúncio multicast dos seus serviços para os pontos de controle. Uma descrição UPnP inclui listagens das ações que podem ser realizadas pelo serviço e das variáveis que representam o estado do serviço em tempo de execução. Os valores das variáveis são atualizados pelo serviço através de mensagens de eventos expressas em XML e formatadas com GENA (*General Event Notification Architecture*) (COHEN; AGGARWAL, 1998).

No caso de o dispositivo possuir uma URL para apresentação, o ponto de controle pode carregar a página desta em um navegador, possibilitando que usuário controle o dispositivo e veja seu estado. Esta característica varia de acordo com os recursos disponíveis na página. Após um ponto de controle obter a descrição de um dispositivo, ele pode enviar a este comandos expressos em XML utilizando SOAP (SNELL; TIDWELL; KULCHENKO, 2002). Em resposta a essas mensagens de controle, o dispositivo envia valores específicos para ações executadas ou códigos de erro. De maneira semelhante, quando um novo ponto de controle é adicionado à rede, ele busca por dispositivos enviando mensagens multicast. Os dispositivos devem responder em unicast para o requisitante.

Em relação a segurança, o modelo utilizado no UPnP provê mecanismos para proteger as mensagens de controle e respostas SOAP, contemplando aspectos como identificação, integridade, autenticação, autorização e privacidade (ELLISON, 2003). É importante destacar que

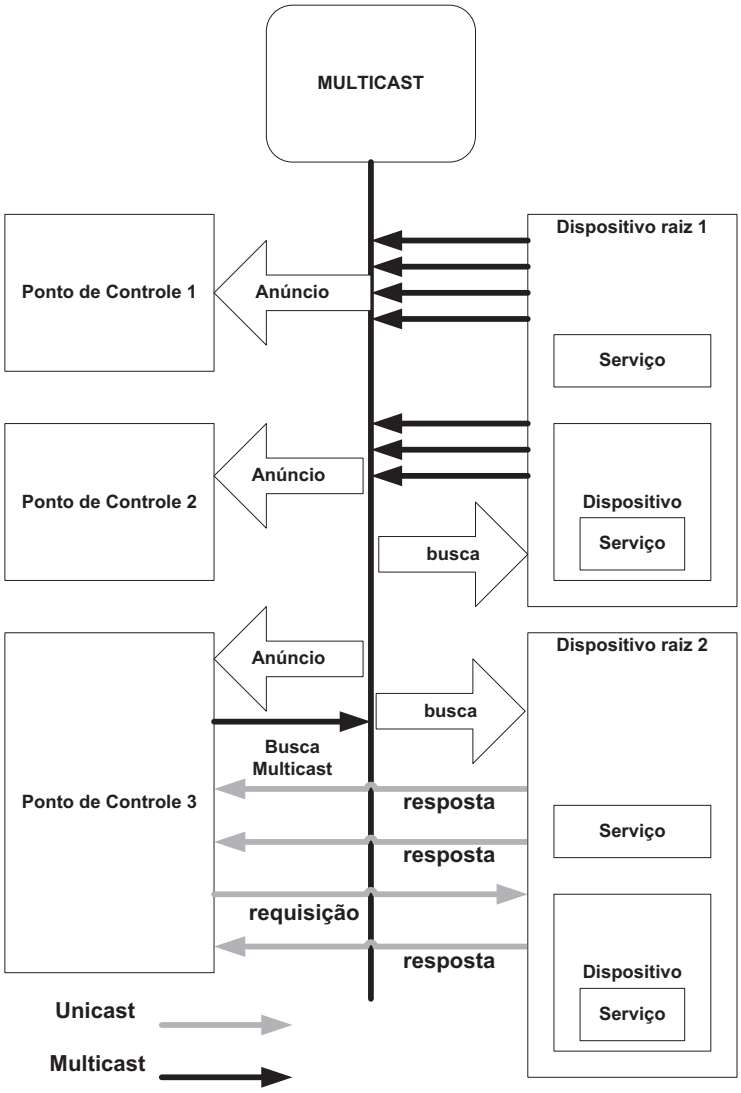

Figura 11: Descoberta de serviços no UPnP

os pontos de controle descobrem dispositivos controlados, mas o contrário não ocorre. O processo realizado para garantir a segurança das mensagens SOAP baseia- se em dados como a lista de proprietários do dispositivo, a lista de acesso, certificados armazenados no dispositivo e certificados enviados pelo ponto de controle como parte da mensagem.

As mensagens recebidas são verificadas e o remetente é identificado pelo hash da sua chave pública. Porém, é possível que uma mensagem não esteja assinada e neste caso, um numero reduzido de ações do dispositivo fica disponível. Se a mensagem está assinada, a assinatura é verificada e é retornado um erro se a mesma não coincide. Nos casos em que é necessário autorização, as informações (como por exemplo, certificados) são solicitadas neste momento e a ação requisitada é executada se o remetente possui privilégios suficientes, gerando uma mensagem de erro caso contrário.

Um dispositivo protegido pode ser descoberto no modo normal ou seguro, diferindo dos demais pelo fato de oferecer um serviço específico, o *DeviceSecurity*. A comunicação entre o proprietário e o dispositivo é feita através de uma interface, que é um componente que atua como dispositivo e ponto de controle – o *Security Console* (SC). O processo seguro de descoberta tem a função de proteger a comunicação entre proprietários de dispositivos e dispositivos e entre dispositivos. Para garantir a privacidade na troca de informações entre estas partes são utilizadas chaves de sessão, que são trocadas através de um protocolo específico do UPnP.

3.4 Zeroconf

O grupo Zeroconf (*Zero Configuration Networking*) (CHESHIRE; STEINBERG, 2006) foi criado pelo IETF em 1999 com o objetivo de definir um conjunto de tecnologias para possibilitar a criação de redes IP totalmente funcionais de forma automática, isto é, sem a necessidade de qualquer intervenção humana e/ou uso de servidores dedicados.

Atualmente, estas tecnologias fornecem mecanismos que permitem a autoconfiguração de endereço IP no enlace local, resolução de nomes e descoberta de serviços de forma descentralizada (ENGELSTAD; EGELAND, 2006).

A autoconfiguração de IP e especialmente útil em pequenas redes desprovidas de infraestrutura ou nas situações em que um servidor de DHCP falha. Nestas condições, a própria máquina deve escolher um IP da faixa 169.254/16, reservada para este propósito e enviar três requisições ARP cujo alvo deve ser o endereço desejado. Se nenhuma resposta é recebida, deve ser enviado um ARP adicional com a fonte e o alvo contendo o IP desejado. Neste momento, a máquina assume o endereço escolhido.

O mDNS (*Multicast DNS*) (CHESHIRE; KROCHMAL, 2011a) e o DNS-SD (*DNS based Service Discovery*) (CHESHIRE; KROCHMAL, 2011b) fazem uso de operações e tipos de dados tradicionais do DNS (MOCKAPETRIS, 1987) e, combinados, permitem a descoberta e anúncio de serviços em uma sub-rede. Estas tecnologias possibilitam a procura por tipos de serviços específicos na rede e que o acesso aos serviços seja feito através de nomes amigáveis.

As eventuais mudanças nos endereços IP e/ou portas onde os serviços são disponibilizados acontecem de forma transparente para o usuário. Para diferenciar os nomes utilizados localmente dos nomes de domínios existentes é utilizado o domínio .local como um pseudo domínio de primeiro nível (TLD). Assim como acontece com os endereços IP da faixa 169.254/24, os nomes do domínio .local fazem sentido apenas no enlace local.

Cada tipo de serviço possui uma identificação como tipo de serviço protocolo de transporte, como por exemplo, ftp._tcp para um servidor de FTP, cuja busca poderia retornar uma resposta como Vendas._ftp._tcp. Ainda é possível refinar as buscas utilizando subtipos, indicados por subtipo._sub._tipo de serviço._protocolo de transporte. Porém, o uso de subtipos nos protocolos é opcional (CHESHIRE; STEINBERG, 2006). Nas máquinas com mDNS é executado em *background* o *mDNS Responder*. Este processo se registra nos grupos multicast, responde as solicitações dos outros hosts, publica seus registros e fica responsável pela cache do mDNS.

Os pacotes contendo requisições contém também uma listagem de possíveis respostas as solicitações e são enviados em intervalos que crescem exponencialmente, podendo chegar a uma hora. Desta forma, se não ocorrem mudanças na rede, nenhum participante fornecerá respostas e a relação de serviços disponíveis é mantida atualizada por que cada novo participante da rede anuncia gratuitamente os seus serviços.

O formato da mensagem mDNS é muito semelhante ao da mensagem DNS unicast. Abaixo, são relacionadas as principais diferenças entre os formatos:

- Os pacotes mDNS podem conter mais de uma solicitação, ao contrário do que acontece com o DNS tradicional;
- O tamanho dos pacotes mDNS pode chegar ate 9000 bytes, enquanto que no DNS unicast o limite e de 512 bytes;
- O mDNS utiliza a porta UDP 5353, em vez da porta 53;

Para anunciar seus serviços, o provedor envia respostas sem que tenham sido feitas requisições prévias, verificando antes se o nome escolhido para a instancia do serviço não esta em uso por outro provedor. Considerando o exemplo do servidor FTP, como não podem existir duas instancias Vendas._ftp._tcp, uma vez detectado a existência de conflito poderia ser escolhido um nome alternativo, como Vendas1._ftp._tcp. A Figura 12 mostra o formato dos pacotes para descoberta e anúncio de serviços.

Com o uso de DNS-SD, as informações do serviço como nome, protocolo de transporte, nome da máquina provedora, endereço IP e porta devem ser disponibilizados no registro SRV, enquanto que informações adicionais podem constar no registro TXT, como por exemplo, a versão de um determinado protocolo.

É importante destacar que o DNS-SD funciona bem com mDNS e com DNS tradicional. Porém, como o tráfego mDNS é descartado pelos roteadores, para tornar possível a descoberta de serviços entre diversas sub-redes (e na Internet) deve ser utilizado DNS unicast. Claramente,

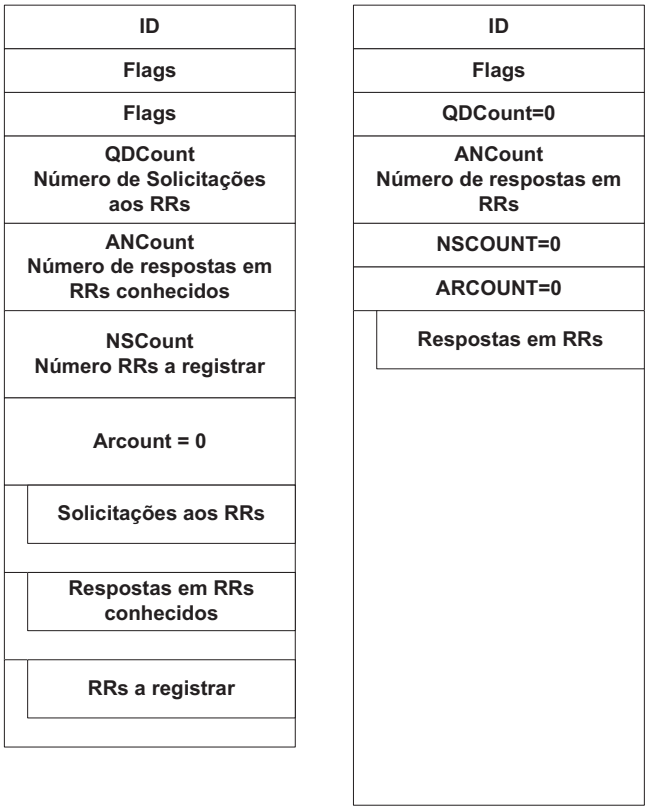

Figura 12: Descoberta e anúncio com mDNS: Pacote com requisições e pacote com respostas enviadas sem requisição prévia, para anúncio de serviços

neste caso é necessário ter uma infraestrutura de rede funcional e além disso, o host deve "saber" em quais domínios deve procurar por serviços, entre os milhares de domínios existentes. Estas e outras questões relacionadas a descoberta de serviços na Internet são tratadas com detalhes em (CHESHIRE; STEINBERG, 2006).

Entre as ferramentas que utilizam mDNS e DNS-SD, podem ser citados o Bonjour (APPLE, 2006) e o Avahi[2], projetado para o sistema operacional Linux. Ambos tiveram seus códigos-fonte disponibilizados pelos autores e tem sido bastante utilizados.

A principal desvantagem do mDNS está relacionada a segurança: devido ao fato de que este protocolo foi projetado para redes locais, nenhum mecanismo de segurança foi desenvolvido.

Uma possibilidade considerada pelos desenvolvedores de ferramentas que utilizam esta tecnologia seria incorporar o uso do DNSSEC (ARENDS et al., 2005) – que foi projetado para o DNS tradicional – ao mDNS, fazendo com que o tráfego não criptografado fosse descartado pelos hosts. Porém, o DNSSEC não é utilizado em nenhuma das implementações de mDNS existentes atualmente. Esta tecnologia é descrita na próxima seção.

[2]http://avahi.org

3.5 DNSSEC

O DNSSEC (ARENDS et al., 2005) é um conjunto de extensões propostas para o protocolo o DNS com objetivo de torná-lo seguro. As extensões definem os métodos para certificação da origem dos dados de zonas e verificação de integridade através de criptografia de chaves públicas.

No DNSSEC, os servidores-raiz fornecem dados para validação aos domínios de primeiro nível, que por sua vez fornecem informações para os domínios de segundo nível, etc. A certificação da origem dos dados é obtida através do uso de assinatura do hash dos dados com a chave privada pertencente a zona, uma vez que os dados não são confidenciais e por este motivo, não necessitam ser criptografados.

Estas assinaturas são acrescentadas aos dados que elas autenticam como registros RRSIG no arquivo de zona. Cada RR de uma zona terá associado um RRSIG e um único par de chaves por zona é utilizado. De posse da chave pública de uma zona, um servidor DNS seguro poderá verificar a validade dos dados recebidos.

O RRSIG contém as seguintes informações:

- Tipo de conjunto de registros que é assinado;
- Algoritmo de assinatura utilizado;
- TTL do conjunto de registros;
- Momento em que os dados foram assinados;
- Momento de expiração da assinatura;
- Identificador de chaves;
- Nome do assinante;
- Asinatura digital.

A assinatura dos RRs das respostas com RRSIGs não resolve o problema das respostas negativas, fornecidas por um servidor quando um nome ou tipo procurado não existem. Isto é resolvido com a introdução do registro NSEC. Este registro carrega a informação de que o nome procurado não existe, o nome mais próximo imediatamente anterior e os tipos (como por exemplo, A ou MX) associados a ele.

3.6 PrudentExposure

Em PrudentExposure (ZHU; MUTKA; NI, 2006) é apresentado um modelo focado em ambientes pervasivos, no qual as informações dos usuários e dos serviços são expostas de forma gradual antes do início do processo de descoberta de serviços.

Na arquitetura proposta, a parte requisitante é formada por clientes e *User Agents — UAs*, enquanto que a parte provedora é composta por diretórios e serviços. Os UAs tem a função de gerenciar as credenciais dos usuários, associando em uma tabela as identidades de domínio com os respectivos domínios, métodos de autenticação e informações para autenticação (como nomes de usuários e senhas, por exemplo). Os diretórios, por sua vez, autenticam clientes e mantem listas de controle de acesso.

Assim como os UAs, os diretórios podem rodar em dispositivos como PCs, servidores e celulares, entre outros. Um diretório armazena apenas serviços pertencentes a um mesmo proprietário e um serviço só se anuncia ao diretório do seu proprietário. Além disso, as informações armazenadas não são anunciadas periodicamente: na presença de uma mensagem de descoberta, o diretório verifica se o remetente e um usuário válido para o serviço e responde em caso afirmativo, do contrário, permanece em silêncio. O protótipo utiliza tecnologias como IEEE 802.11b e UDP multicast, mas pode ser utilizado também em redes cabeadas.

O processo de descoberta é composto pelas seguintes etapas: busca por domínios disponíveis, autenticação, seleção do serviço, troca de chaves e invocação do serviço. Inicialmente, um UA procura por domínios disponíveis e quando encontra, seleciona as credenciais corretas para autenticação. Após, UA e cliente solicitam informações relacionadas aos serviços. Quando recebem respostas, o cliente deve então selecionar um serviço. Na etapa seguinte, uma chave criptográfica é entregue ao serviço pelo diretório e ao cliente pelo UA. Neste momento, o cliente já pode utilizar o serviço.

No caso do registro de um serviço devem ser cumpridas apenas duas etapas: a busca pelo domínio e o registro propriamente dito. Quando um serviço encontra um domínio ele envia uma mensagem de registro utilizando um canal seguro.

Para que a busca por domínios ocorra de forma segura, os diretórios e UAs falam *code words*, de modo que seja estabelecida uma relação de confiança entre as partes. As code words são expressas na forma de *Bloom filters* (BRODER; MITZENMACHER, 2002) e são obtidas pela aplicação de funções de hash a identidade de domínio – que é um segredo compartilhado entre o domínio e seus usuários — e a um parâmetro variável no tempo, que é composto por um número aleatório e um time-stamp, tornando as code words dinâmicas. Dada a possibilidade de múltiplos domínios coexistirem em um mesmo lugar e para possibilitar a descoberta em diferentes locais, um UA pode falar muitas code words (para procurar por diversos domínios).

Os *Bloom filters* são representados como arrays de bits, com tamanho igual ao range das funções de hash utilizadas — neste caso MD5, SHA-1 e RIPEMD-160. O array é inicializado com zeros. As funções de hash são aplicadas ao(s) elemento(s) em questão e utilizando-se um resultado do hash como índice, um bit é setado no array. Um mesmo bit pode ser setado pelo uso de diferentes elementos ou pelo uso de diferentes funções de hash. Neste caso, aplicam-se as funções de hash a identidade de domínio e ao parâmetro variável no tempo, que também é enviado na mensagem de descoberta. A Figura 13 mostra um exemplo de *Bloom filter*.

Na figura o array é inicializado com zeros. Em seguida, são aplicadas diferentes funções

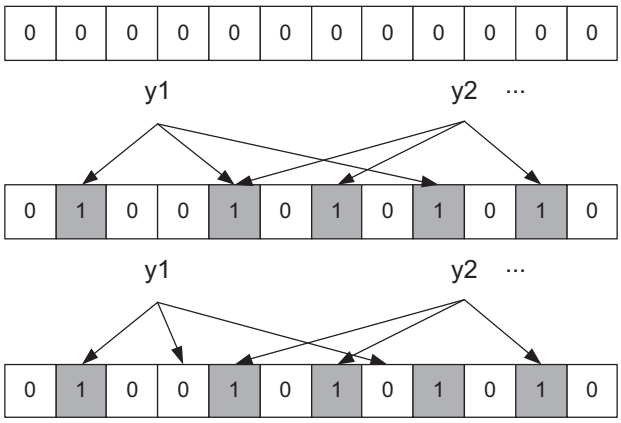

Figura 13: Bloom Filter

de hash aos elementos X1 e X2, setando os bits correspondentes. Para verificar os elementos Y1 e Y2, aplicam-se a estes as funções de hash. Para Y1, foi encontrado 0 em uma posição do array que deveria conter 1, indicando falha. Já Y2 constitui um caso de correspondência ou falso positivo.

O uso dos *Bloom filters* possibilita o envio de diversas *code words* em um mesmo pacote, permitindo que sejam enviadas requisições para diferentes domínios. Para o destinatário verificar se um usuário é legítimo, ele aplica as funções de hash a estes elementos utiliza os resultados como índices. Cada zero encontrado na posição correspondente significa usuário não–legítimo. Do contrário, o usuário é considerado válido e uma mensagem deve ser enviada, contendo o bit setado inicialmente no array e um bit setado pelo uso de outra função de hash (aplicada aos mesmos parâmetros).

Se houver necessidade, uma nova rodada é realizada. Senão, ocorre uma troca de chave para acesso as informações dos serviços disponíveis. Esta chave é valida para uma sessão, sendo que a mesma é repassada ao cliente e ao serviço através do UA e do diretório, respectivamente. Após a seleção do serviço, uma nova chave é estabelecida, desta vez para acesso ao serviço.

A principal desvantagem deste sistema é que existe uma pequena possibilidade de falso positivo, podendo ocorrer em alguns casos a exposição parcial de informações sigilosas. Porém, todas as vezes que um usuário legítimo tenta descobrir um serviço ele consegue. Para poder coexistir com outros modelos nos quais não são utilizados meios para autenticação, devem ser feitas modificações, como a reserva de bits nos filtros para representar domínios onde não é necessário autenticação para descoberta de serviços. Um outro problema não tratado neste trabalho é o de como suspender o acesso de um usuário a um determinado serviço, uma vez que a identidade de domínio é compartilhada com todos os usuários, criando a necessidade de distribuir um novo segredo.

3.7 Private and Secure Service Discovery via Progressive and Probabilistic Exposure

O modelo de (ZHU et al., 2007) propõe uma exposição gradativa dos dados do cliente e do servidor (identidade e informações sobre os serviços, respectivamente) de modo que seja estabelecida uma relação de confiança entre as partes após múltiplas rodadas de comunicação.

No sistema proposto, é possível evitar o tráfego de informações desnecessárias nos casos em que uma das partes não atenda aos requisitos para completar o processo de descoberta. Já em PrudentExposure, dos mesmos autores, ainda que o usuário não possua privilégios suficientes para acessar os serviços ou o provedor não ofereça os serviço requisitado, pode ocorrer exposição de dados como a identidade do servidor e/ou do cliente e o serviço procurado.

Neste sistema, parte-se da premissa de que usuário e provedor tenham compartilhado previamente um segredo para a descoberta e ainda, que as partes compreendem dados de identificação parciais trocados de forma criptografada. Um usuário é considerado legítimo se compartilha um segredo com o provedor e se tem privilégios suficientes para acessar um determinado serviço. Já o provedor é considerado legítimo se compartilha um segredo com o usuário e possui o serviço requisitado.

A exposição progressiva de dados ocorre da seguinte forma: os clientes e provedores expõem dados criptografados (alguns bits de uma *code word*) em cada rodada da descoberta. Como no trabalho anterior, uma code word é gerada a partir da aplicação de uma função de hash a um segredo compartilhado entre um usuário e um provedor de serviços e a um parâmetro variável no tempo. O usuário envia alguns bits de uma *code word* e o provedor verifica a mensagem, permanecendo em silêncio caso não reconheça a informação recebida. De outra forma, ele envia outros bits da code word para que o cliente verifique.

Em cada etapa, os clientes enviam alguma informação sobre a sua identidade e o serviço buscado, enquanto que os provedores divulgam parcialmente a sua identidade e os serviços que oferece. Em todas as rodadas servidor e cliente verificam os dados recebidos, suspendendo a comunicação caso ocorra algum erro. Se nenhum erro acontece, em algum ponto as partes passam a confiar uma na outra, ou seja, acreditam que o outro é legítimo com alta probabilidade e estabelecem uma conexão para uso do serviço. Para tanto, o protótipo utiliza UDP nas primeiras mensagens e em seguida TCP, com o objetivo de garantir a entrega das informações. De acordo com os autores, o sistema apresenta desempenho satisfatório na maioria dos casos se executado em dispositivos como PDAs.

Como desvantagem, não é possível realizar descoberta com base nos atributos dos serviços. O sistema também apresenta uma limitação para os casos em que existe uma grande quantidade de usuários: a ocorrência de falsos positivos aumenta de forma significativa, principalmente nas primeiras mensagens. Além disso, os cálculos das probabilidades são feitos com base nos históricos mantidos pelas partes. Como consequência natural, um provedor que muda de ambiente com frequência irá dispor de uma quantidade de dados reduzida para realizar seus cálculos.

Um outro aspecto observado pelos autores é que o processo de descoberta é o mesmo para todos os tipos de serviço, ainda que para determinadas aplicações a exposição de dados seja mais crítica do que para outras.

3.8 Protocolos para descoberta de serviços em redes *ad hoc*

Existe uma quantidade significativa de trabalhos focados exclusivamente em redes *ad hoc*. Os modelos desenvolvidos para estas redes tratam de modo geral, questões complexas relacionadas ao estabelecimento da comunicação neste tipo de ambiente, como por exemplo o poder de processamento reduzido em alguns dispositivos, a necessidade de economizar energia e a distância entre os equipamentos.

Embora estas tecnologias sejam destinadas a ambientes diferentes daqueles para o qual o Sec-SD foi pensado, faz-se necessário descrever algumas propostas focadas em redes *ad hoc* devido a crescente utilização deste tipo de rede para os mais diversos fins, como por exemplo, dispositivos para monitoração da saúde de pacientes, ambientes de desastres naturais e redes de sensores, entre tantos outros. Abaixo, são comentadas algumas propostas que visam atender estas necessidades.

O Bluetooth (IEEE, 2005) é uma tecnologia baseada em RF de curto alcance e largura de banda relativamente baixa, bastante utilizada em dispositivos como telefones celulares e PDAs. A descoberta de dispositivos na rede e a descoberta de serviços em um determinado dispositivo ocorrem em camadas distintas da pilha de protocolos Bluetooth. Para encontrar dispositivos, o cliente envia uma mensagem utilizando o L2CAP (*Logical Link Control and Adaptation Protocol*) e o LMP (*Link Manager Protocol*) e os participantes da rede aptos para descoberta respondem com seus endereços.

Um dispositivo Bluetooth pode não responder por estar em modo de economia de energia ou por motivos de segurança — as informações são criptografadas na camada de enlace, restringindo a comunicação aos peers que compartilham segredos (THOMSEN et al., 2002). No nível superior, o Bluetooth SDP (*Service Discovery Protocol*) (SAIRAM; GUNASEKARAM; REDDY, 2002) provê mecanismos para pesquisar quais serviços estão disponíveis em um determinado dispositivo – que pode atuar como cliente SDP (quando busca serviços em outros componentes da rede), servidor SDP (quando oferece serviços) ou ambos. Este protocolo não especifica métodos de acesso aos serviços. Um servidor SDP mantém uma lista de registros que descrevem as características dos serviços que ele disponibiliza. Cada registro é composto por uma lista de atributos, que por sua vez são representados por pares <identificador, valor>.

O modelo Rescue (JUSZCZYK et al., 2008) é focado em redes *mesh* e ambientes totalmente *ad hoc*, isto é, sem nenhuma infraestrutura instalada, sendo voltado para uso de dispositivos móveis em ambientes de desastres naturais, como terremotos e enchentes. Nestes casos a infraestrutura instalada pode não estar operacional ou não ser confiável em razão de diversos fatores. O Rescue utiliza mensagens multicast e protocolos de roteamento como o VRR (*Virtual*

Tabela 1: Comparação entre tecnologias para descoberta de serviços.

Sistema / Característica	Arquitetura de descoberta	Transporte	Escopo	Registro e descoberta	Busca	Segurança
Jini	Híbrida	UDP multicast / TCP	Sub-rede/bridge	Requisição e anúncio	Tipo, ID ou atributo	Autenticação / confidencialidade / integridade
UPnP	Distribuída	HTTP unicast e multicast sobre UDP	Sub-rede	Requisição e anúncio	Tipo ou ID	Autenticação /confidencialidade / integridade / controle de acesso
SLP	Centralizada ou distribuída	UDP multicast / TCP	Sub-rede/bridge	Requisição e anúncio	Tipo ou atributo	Autenticação (opcional)
mDNS/DNS-SD	Distribuída	UDP multicast	Sub-rede	Requisição e anúncio	Tipo	DNS-SEC (opcional para autenticação)
PrudentExposure	Centralizada (cliente-agente-diretório-serviço)	UDP multicast	Sub-rede	Registro seguro junto ao diretório/Requisição	ID/senha (hashes)	Autenticação / confidencialidade

Ring Routing) (CAESAR; CASTRO; NIGHTINGALE, 2006) e o OLSR (*Optimized Link State Routing*) (CLAUSEN; JACQUET, 2003) para anúncio dos serviços.

O Konark (HELAL; DESAI; VERMA, 2003) foi uma das primeiras propostas pensadas exclusivamente para redes *ad hoc*, levando em conta a complexidade deste tipo de ambiente. O Konark pode ser utilizado para descoberta de dispositivos físicos, como impressoras ou para serviços de sofware, como comércio eletrônico ou entretenimento. Este protocolo é concebido no nível IP e portanto, independe do meio físico utilizado, que pode ser por exemplo o Bluetooth ou 802.11. Para manter compatibilidade com diversos dispositivos, é utilizado HTTP e SOAP com mensagens em XML para as as requisições e respostas.

O OLSR-mDNS (KREBS; KREMPELS; KUCAY, 2008) é uma proposta para redes *mesh* sem-fio que utiliza *multicast DNS* encapsulado em OLSR — que é um protocolo de roteamento de estado de enlace, possibilitando que as mensagens mDNS, que normalmente são descartadas pelos roteadores possam ser encaminhadas através destes dispositivos. A arquitetura deste protocolo é composta por roteadores sem-fio em topologia *mesh*, *routing* clients, que são dispositivos que executam o algoritmo OLSR e *non-routing clients*, que não executam OLSR mas rodam uma aplicação para anúncio e descoberta de serviços. O modelo utiliza ainda uma cache DNS-SD distribuída entre os roteadores.

3.9 Considerações sobre os modelos apresentados

A seguir são comentadas algumas das vantagens e desvantagens das tecnologias estudadas, sendo destacados os aspectos que são mais relevantes do ponto de vista deste trabalho. A Tabela 1 mostra uma comparação entre as tecnologias apresentadas, com excessão daquelas que são focadas em redes *ad-hoc*, pelo fato de os ambientes para os quais elas foram projetadas diferirem muito daqueles para o qual o Sec-SD foi pensado, como já foi comentado anteriormente.

O modelo PrudentExposure tem como principal vantagem o fato de que um usuário pode procurar por diversos serviços sem que suas credenciais sejam enviadas de forma direta. Isso

pode ser obtido com o uso dos Bloom filters, que são arrays de bits cujas posições são setadas a partir do resultado de funções de hash aplicadas sobre as credenciais em conjunto com um parâmetro variável no tempo. O uso deste parâmetro faz com que a repetição de mensagens capturadas por entidades mal-intencionadas não tenha nenhum efeito.

Após a negociação inicial são estabelecidas chaves de sessão para criptografar as solicitações de informações referentes aos serviços e as respectivas respostas. Como desvantagem, existe a possibilidade de falsos-positivos nas buscas por serviços. Além disso, este modelo centraliza o gerenciamento dos serviços e também das credenciais dos usuários, visto que o mesmo é focado em ambientes pervasivos, onde é comum a existência de dispositivos heterogêneos.

O Jini utiliza na descoberta segura de dispositivos mecanismos para a criptografia, autenticação e verificação da integridade dos dados. No projeto de aplicações com Jini, é possível escolher entre diferentes protocolos para prover estes requisitos de segurança. O Jini é um exemplo de tecnologia híbrida (que pode ou não empregar centralização no gerenciamento dos serviços através do uso dos serviços de consulta, os *(Lookup Services)*. No que diz respeito a autenticação de usuários esta é feita de forma centralizada, como por exemplo, por um servidor Kerberos.

O UPnP pode ser utilizado em redes com ou sem infraestrutura, utilizando endereços IP de uma faixa especial no segundo caso. Os dispositivos UPnP possuem papéis bem definidos: pontos de controle ou dispositivos controlados. No que diz respeito ao modelo de segurança empregado, são oferecidos mecanismos para identificação, verificação de integridade, autenticação, autorização e privacidade através do uso de lista de proprietários dos dispositivos, listas de acesso e certificados. Estes procedimentos tem como objetivo proteger a comunicação entre proprietários de dispositivos e dispositivos e entre dispositivos. A comunicação entre usuários humanos e dispositivos é feita através de uma interface que por sua vez deve ser associada aos dispositivos corretos. A privacidade na troca de informações entre estas partes é conseguida com o uso de chaves de sessão, que são trocadas através de um protocolo específico do UPnP.

As tecnologias mDNS e DNS-SD utilizam nomes amigáveis para identificar os serviços disponíveis na rede, podendo incluir também informações adicionais no registro TXT do DNS, que podem auxiliar o usuário na escolha dos serviços, como por exemplo, se uma impressora imprime colorido ou preto e branco. Como o acesso é feito através do nome, as mudanças nos endereços IP e/ou portas onde os serviços são disponibilizados ocorrem de forma transparente para o usuário. Um outro aspecto positivo destas tecnologias é a redução significativa do tráfego multicast na rede obtida através do envio de respostas conhecidas juntamente com as solicitações. Além disso, a descoberta de serviços com DNS-SD funciona normalmente com o DNS tradicional, possibilitando seu uso em redes como a Internet. Como desvantagem, o mDNS e o DNS-SD não empregam mecanismos de segurança.

O uso do DNSSEC em conjunto com mDNS e DNS-SD iria prover meios para verificação da origem dos dados e da integridade dos mesmos através da assinatura das mensagens de resposta. Contudo, essas informações podem ser verificadas por qualquer entidade que conheça a chave

pública do remetente. Essa característica se deve ao fato de que o DNSSEC foi projetado para o DNS, cujas mensagens são de domínio público e não necessitam portanto ser criptografadas.

Entre os protocolos destinados a redes *ad hoc* que foram estudados, somente o Bluetooth contempla mecanismos de segurança na descoberta de dispositivos. Contudo, tal mecanismo é empregado em nível de enlace, sendo a segurança garantida pelo fato de que a comunicação fica restrita aos dispositivos que compartilham segredos. A descoberta de serviços é tratada em uma camada superior, pelo Bluetooth SDP *(Service Discovery Protocol)*.

Com o objetivo de permitir o aproveitamento de características já existentes em protocolos para descoberta de serviços, possibilitando desta forma agregar tais mecanismos no Sec-SD, foram avaliados diversos aspectos em cada um dos protocolos estudados. Os mais importantes para o escopo deste trabalho, de acordo com as premissas definidas para o Sec-SD, são relacionados abaixo:

- A arquitetura utilizada para descoberta (centralizada, distribuída ou híbrida);
- O escopo de funcionamento;
- Os critérios utilizados na descoberta, como por exemplo tipos ou identificadores para os serviços;
- Os mecanismos de segurança (quando empregados);
- Quando empregados mecanismos de segurança, o fato de o protocolo em questão empregar soluções com ou sem centralização.
- O tráfego na rede levando-se em conta as mensagens trocadas na descoberta segura de serviços.

Com base nessa avaliação, foram identificados diversos aspectos desejáveis no funcionamento do Sec-SD. Porém, é importante destacar que tais características são encontradas isoladamente nos protocolos estudados.

Conforme foi descrito anteriormente, o Sec-SD tem como requisito que todos os nodos da rede possam atuar como cliente e servidor simultaneamente, que a descoberta ocorra de forma segura e totalmente descentralizada em ambientes locais e que a existência de provedores de serviço redundantes seja tratada de a fim de ser transparente para o usuário humano. Foi constatado que nenhum dos protocolos estudados possui este conjunto de características.

Neste ponto, é fundamental destacar que para a realização da descoberta de serviços e da autenticação de usuários serem realizadas sem o emprego de diretórios é fundamental que cada dispositivo da rede seja capaz de gerenciar os serviços que ele oferece bem como as credenciais para acesso aos serviços que ele possui privilégios para utilizar.

Desta forma, todos os dispositivos devem ser capazes de desempenhar as funções necessárias para verificar as identidades dos seus potenciais clientes e também para se autenticar junto aos

seus provedores. Além disso, o protocolo também deve tratar a existência de provedores de serviço redundantes.

Como nenhum dos modelos descritos reúne tais características, assim como outros existentes na literatura, faz-se necessário combinar aspectos encontrados em tecnologias já existentes com o desenvolvimento de mecanismos específicos para que seja possível obter o comportamento desejado no Sec-SD, o que representa o objetivo principal deste trabalho.

Entre os mecanismos para comunicação e métodos de segurança existentes (e largamente utilizados) que são empregados nos protocolos de descoberta de serviços estudados e que também são utilizados no Sec-SD, destacam-se o envio de mensagens multicast, o uso de identificadores para os serviços e os métodos para autenticação baseados em criptografia simétrica e assimétrica. Uma descrição detalhada do modelo Sec-SD é apresentada no próximo capítulo.

4 SEC-SD: DESCOBERTA DE SERVIÇOS SEGURA E DESCENTRALIZADA EM AMBIENTES LAN

As tecnologias para descoberta de serviços visam simplificar para os usuários as tarefas de encontrar e utilizar recursos em uma rede, minimizando a necessidade de auxílio técnico na ocorrência de mudanças. Ao mesmo tempo, essas aplicações possibilitam a cooperação entre dispositivos e serviços, reduzindo e facilitando as tarefas administrativas.

Diante da crescente preocupação em integrar mecanismos de segurança nos protocolos de descoberta de serviços, com o objetivo de atender os requisitos necessários em diversos ambientes, tais como redes domésticas, pequenos escritórios, empresas e universidades, em redes cabeadas ou *wireless*, o Sec-SD (*Secure Service Discovery Protocol*) é uma proposta para descoberta segura de serviços em redes locais com arquitetura totalmente descentralizada.

Com uso do Sec-SD, cada dispositivo pode atuar ao mesmo tempo como cliente e provedor de serviços sem a necessidade de utilizar diretórios para registrar os seus serviços e/ou procurar por serviços disponíveis. O protocolo também trata a existência de provedores redundantes para um serviço no ambiente, fazendo com que a presença destes seja transparente para os usuários humanos.

O Sec-SD utiliza criptografia e autenticação para tornar o seguro o processo de descoberta de serviços e minimiza a exposição de informações confidenciais por parte do cliente e do provedor de serviços. Além disso, são empregados mecanismos para reduzir o tráfego de mensagens de descoberta, visando tornar o sistema escalável e adequado também para ambientes com grande número de máquinas.

Nas próximas seções são apresentados os requisitos de funcionamento do Sec-SD, o detalhamento do protocolo e os formatos das mensagens.

4.1 O modelo Sec-SD

O Sec-SD é um *four-way handshake* que tem como objetivo prover uma chave criptográfica a partir de uma senha. A motivação para isso é que a senha é um mecanismo fácil para usuários humanos mas fraco para segurança. Assim, a senha é utilizada somente na interação inicial, enquanto que a chave, que é criptograficamente mais forte, é utilizada para a obtenção de informações relacionadas aos serviços e acesso aos mesmos. Para reforçar a segurança, a chave é renovada periodicamente.

As entidades do Sec-SD são descritas abaixo:

Serviço É um recurso presente na rede, que pode ser por exemplo, um protocolo suportado por um software ou dispositivo, tal como SSH, FTP, HTTP, SIP, etc.

Provedor de serviços É uma entidade que disponibiliza um ou mais serviços, como por exemplo, impressoras, servidores de arquivos, um computador pessoal que possua um servidor

de SSH e/ou um servidor Web, centrais telefônicas, *smartphones*, etc.

Proprietário É a pessoa responsável por gerenciar o serviço. Considerando os exemplos acima, o administrador da rede poderia ser o proprietário dos serviços de impressão e de compartilhamento de arquivos, enquanto que os serviços disponibilizados pelo *smartphone* seriam administrados pelo dono do dispositivo.

Usuário/cliente É a entidade que busca por serviços na rede e acessa os mesmos. Pode ser um usuário humano, um software, um dispositivo ou um serviço.

As principais etapas para descoberta de serviços através do Sec-SD são: *busca por serviços*, *autenticação*, *solicitação de chave criptográfica*, *envio da chave* e por último, *solicitação e envio de informações sobre os serviços*. Primeiramente, um cliente requisita um serviço. Se o provedor estiver disponível, ele verifica as informações fornecidas pelo cliente e responde se as mesmas estiverem corretas. O cliente, por sua vez, verifica a resposta e solicita uma chave criptográfica. Se a solicitação contiver as informações corretas, o provedor envia a chave. Neste momento, o cliente pode utilizar a chave para solicitar informações relacionadas ao serviço e então, acessar o mesmo.

Com estas etapas, o Sec-SD não oferece métodos para acesso aos serviços, mas garante que somente os clientes legítimos poderão obter informações sobre os serviços disponíveis e com isso, acessá-los. A chave obtida ao final da negociação entre cliente e provedor *deve* ser utilizada para a aquisição de dados relacionados ao serviço, *podendo* também ser usada para acesso ao mesmo (ex. para criptografar mensagens enviadas na transferência de arquivos entre duas máquinas). Porém, o Sec-SD não especifica métodos para integração com outros softwares, desde que sejam observadas as definições do protocolo, tais como o tempo de validade das chaves fornecidas pelos provedores.

A interação entre as entidades do protocolo em cada uma das etapas citadas acima é explicada em detalhes nas seções seguir.

4.1.1 Gerenciamento de serviços

Para permitir a descoberta de serviços via Sec-SD, o proprietário de um serviço (por exemplo, um administrador de rede) atribui um identificador e uma senha para cada serviço e para permitir que um determinado grupo de clientes tenham acesso a um serviço específico, ele repassa ao grupo estas informações.

Um cliente deve possuir as credenciais — neste caso, o identificador e a senha — de cada serviço que ele pode acessar, ainda que alguns (ou muitos) dos serviços pertençam a um mesmo proprietário e/ou sejam disponibilizados pelo mesmo provedor. Desta forma, o Sec-SD permite que a autenticação seja realizada de forma individual para cada serviço, ou seja, considerando as credenciais fornecidas pelo cliente e o serviço requisitado.

Em diversos ambientes não é desejável que um cliente de posse de uma única senha possa acessar todos os serviços pertencentes a um mesmo proprietário, sejam eles disponibilizados por um único dispositivo ou ainda, providos por um conjunto de máquinas presentes na rede.

Como exemplo, podemos considerar uma máquina que atua como servidor de arquivos e servidor Web em uma empresa e que esteja sob responsabilidade do administrador da rede. É natural que as páginas Web estejam disponíveis a uma grande quantidade de usuários, enquanto que os arquivos sejam de acesso restrito a um grupo de funcionários. Assim, não é recomendável que exista uma senha única para a máquina que disponibiliza os serviços, e sim uma senha para o servidor Web e outra para o servidor de arquivos.

Pelo fato de utilizar credenciais diferentes para cada serviço, o Sec-SD permite que um mesmo dispositivo seja provedor de serviços pertencentes a diferentes proprietários. Considerando o uso do Sec-SD em uma empresa, é possível que as estações de trabalho ofereçam alguns serviços que sejam controlados pelo administrador da rede e outros que estejam aos cuidados do funcionário que utiliza a máquina. Neste caso, a máquina pertencente a um gerente pode disponibilizar um software para gerenciamento de projetos com interface Web a todos os integrantes do seu departamento e alguns outros interessados, deixando o respectivo serviço sob responsabilidade do administrador da rede. Por outro lado, relatórios de avaliação de funcionários do setor possivelmente seriam de responsabilidade deste gerente, que manteria as credenciais em sigilo.

A senhas e os identificadores do serviços constituem as credenciais para acesso aos mesmos, sendo portanto informações confidenciais. Estes dados devem ser armazenados nos dispositivos provedores de forma segura, como por exemplo, em arquivos criptografados. Contudo, o método de armazenamento de credenciais é de escolha dos proprietários dos serviços.

No que diz repeito ao repasse das credenciais aos usuários interessados, é de grande importância que esta ação seja realizada de forma segura. Porém, a escolha desse meio é de responsabilidade dos provedores de serviço. Considerando que o escopo de atuação do Sec-SD é a rede local, a interação direta entre proprietários de serviços e usuários constitui uma alternativa viável.

No momento em que um provedor de serviços ingressa na rede, ele gera uma chave criptográfica distinta para cada um de seus serviços. Esta chave será utilizada para obtenção de informações relacionadas ao serviço e posterior acesso ao mesmo e é denominada *chave de grupo*, visto que é repassada ao grupo de usuários que tem credenciais para um serviço e se autentica junto ao provedor fazendo uso destas.

Um grupo de clientes que possui acesso a um dado serviço ingressa na rede busca pelo mesmo enviando informações obtidas a partir o identificador e a senha deste serviço para obter a respectiva *chave de grupo*. Os provedores também respondem com dados obtidos das credenciais do serviço. Este procedimento é denominado *autenticação para o grupo*, pelo fato de ser realizada com uso das credenciais que são de conhecimento de todo o conjunto de entidades que tem acesso ao serviço e permite que cliente e provedor tenham certeza de que o outro lado

realmente pertence ao respectivo grupo, porém sem identificá-los individualmente.

Um proprietário pode desejar que seus clientes e provedores se autentiquem mutuamente com uso de assinaturas digitais para obtenção da *chave de grupo*. Neste caso, o proprietário deve repassar a chave pública do serviço a estes clientes e solicitar as chaves públicas dos mesmos no momento em que ele fornece o identificador e a senha do serviço. As chaves públicas dos clientes ficam armazenadas no provedor de serviços.

A opção de utilizar *autenticação para o grupo* ou *autenticação de cliente e provedor* deve ser feita pelo proprietário para cada serviço seu, constando em cada dispositivo provedor (ex. em um arquivo de configuração) o devido tipo de autenticação aceito por cada um dos serviços disponíveis. Esses dois tipos de autenticação caracterizam respectivamente os níveis de segurança 1 e 2 do Sec-SD.

As etapas que devem ser cumpridas para autenticação de clientes e obtenção da *chave de grupo* nos dois casos comentados acima, bem como os possíveis cenários para cada tipo de autenticação são explicados detalhadamente nas próximas seções.

No que diz repeito aos procedimentos para compartilhamento de credenciais quando do uso de provedores redundantes, estes serão tratados na seção *Operação em modo de redundância*.

Em relação ao bloqueio de clientes para um determinado serviço, é necessário substituir as credenciais deste serviço para impedir que um cliente continue apto a se autenticar para obter a respectiva chave de grupo.

4.1.2 Operação

O processo de descoberta tem como objetivo principal estabelecer uma relação de confiança entre cliente e provedor de serviços antes da solicitação e fornecimento de informações sobre os mesmos. No momento em que ambas as entidades – cliente e provedor de serviços – se certificam de que o outro lado é confiável, o cliente recebe a chave criptográfica (chave de grupo) para acesso as informações do serviço.

Para garantir a segurança deste procedimento, as credenciais dos clientes não são enviadas através da rede. Ao invés disso, durante a autenticação das partes envolvidas no processo de descoberta são enviados os hashes obtidos a partir da combinação das credenciais associadas aos serviços com *time-stamps*. Assim, as informações enviadas para autenticação são dinâmicas, o que torna mais difícil a ação de entidades mal-intencionadas.

Além de permitir que cada dispositivo possa desempenhar simultaneamente os papéis de cliente e provedor de serviços, o Sec-SD possibilita ainda o emprego de provedores redundantes no ambiente. O uso de redundância para um determinado serviço ocorre de forma transparente para o usuário e não implica em restrições quanto ao número de nodos empregados para este objetivo.

Desta forma, uma entidade pode utilizar a chave de grupo recebida ao final da negociação com o provedor para acessar o serviço, caracterizando a operação em *modo cliente* ou pode

Figura 14: Envio de requisição e resposta no Sec-SD.

também, no caso de ser um provedor, passar a responder requisições e enviar a chave a outros clientes. Este último modo de operação é denominado *modo de redundância* e é explicado com detalhes nas próximas seções.

Abaixo, são descritos os possíveis modos de operação para um dispositivo. É importante destacar que o uso de um ou outro modo é uma escolha para ser feita pelo proprietário/administrador da rede para cada serviço. Um dispositivo pode por exemplo ser ao mesmo tempo provedor do serviço A, cliente do serviço B e provedor redundante do serviço C.

4.1.3 Operação em modo cliente

Este modo de operação é utilizado por uma entidade que deseja se autenticar junto a um provedor e obter as informações necessárias para acessar um determinado serviço.

Neste caso, no momento em que ingressa na rede, um cliente pode dar início ao processo de descoberta de serviços enviando requisições contendo informações relacionadas as suas credenciais. Estes dados são verificados pelos provedores, que respondem somente se os dados recebidos estiverem corretos, permanecendo em silêncio caso contrário. Este processo é mostrado na Figura 14.

Do lado do cliente, se nenhuma resposta for recebida dentro do *timeout* especificado pelo administrador da rede ou na implementação, que deve ser de alguns segundos, ele deve enviar novamente a requisição. Se ainda assim não obtiver resposta, ele assume que o serviço solicitado está indisponível.

Se existe um provedor ativo na rede que responde a requisição, ao final da negociação o cliente obtém a chave de grupo que será utilizada para obtenção de informações sobre o serviço solicitado e acesso a este. Se um serviço é oferecido por diversos provedores redundantes, pode acontecer de um cliente receber mais de uma resposta. Neste caso, o cliente irá considerar uma resposta somente (a que tiver o menor time-stamp) e descartar as demais. As mensagens trocadas entre as partes envolvidas na descoberta são descritas nas seções a seguir.

4.1.4 Operação em modo provedor

Neste modo de operação, o dispositivo entra na rede e gera uma chave de grupo diferente para cada um de seus serviços. Cumprida esta etapa, ele deve aguardar em silêncio por requisições de clientes.

Para evitar exposição desnecessária de informações confidenciais, os provedores não enviam anúncios espontâneos da sua presença ou dos serviços disponíveis. Ao invés disso, eles aguardam o recebimento de requisições e na presença das mesmas, verificam se as credenciais recebidas do cliente são válidas para algum de seus serviços, respondendo em caso afirmativo. Se as credenciais são inválidas, o provedor deve descartar silenciosamente a mensagem.

Quando um provedor envia a chave a um cliente ele envia o tempo pelo qual a chave ainda será válida e determina também o momento exato em que o cliente deverá requisitar a nova chave. Este procedimento é realizado com o intuito de tornar o Sec-SD escalável, para que o mesmo possa ser utilizado também em redes com muitas máquinas. Desta forma, evita-se que todas as máquinas enviem solicitações ao mesmo tempo, porém garantindo que todos os clientes irão obter a nova chave antes da expiração da atual.

4.1.5 Operação em modo de redundância

O Sec-SD suporta a existência de provedores de serviço redundantes, sendo esta empregada com o intuito de assegurar a disponibilidade de serviços considerados essenciais. Neste caso, o protocolo trata a redundância de servidores de modo a torná-la transparente para o usuário, que na presença de um ou mais provedores para um dado serviço deve visualizar um único serviço na rede.

Os serviços no Sec-SD são caracterizados pelos seus identificadores, que constituem uma informação sigilosa, juntamente com a senha. Serviços redundantes possuem o mesmo identificador e a mesma senha. O dispositivo que é provedor redundante e ingressa na rede se comporta do mesmo modo que um cliente, tendo como objetivo obter a chave de grupo. O provedor que recebe a requisição também trata a requisição como faria usualmente para qualquer cliente. A principal diferença do ponto de vista do provedor redundante que acaba de entrar na rede é que ao final da negociação, de posse da chave de grupo, ele passará também a responder requisições.

Na presença de provedores redundantes não é necessária nenhuma ação adicional por parte de um usuário ou de um dispositivo cliente durante o processo de descoberta de um serviço que possui mais de um provedor. Todo o controle é realizado pelo(s) provedor(es). Além disso, não é aplicada nenhuma restrição quanto ao número de nodos redundantes empregados.

Em caso de falha de um ou mais provedores redundantes, o serviço continua aparecendo como disponível para o usuário enquanto houver pelo menos um provedor ativo na rede. A chave de grupo adquirida inicialmente pelos clientes é válida junto a todo o conjunto de provedores redundantes de um dado serviço e dessa forma, continua válida até que seja atingido o

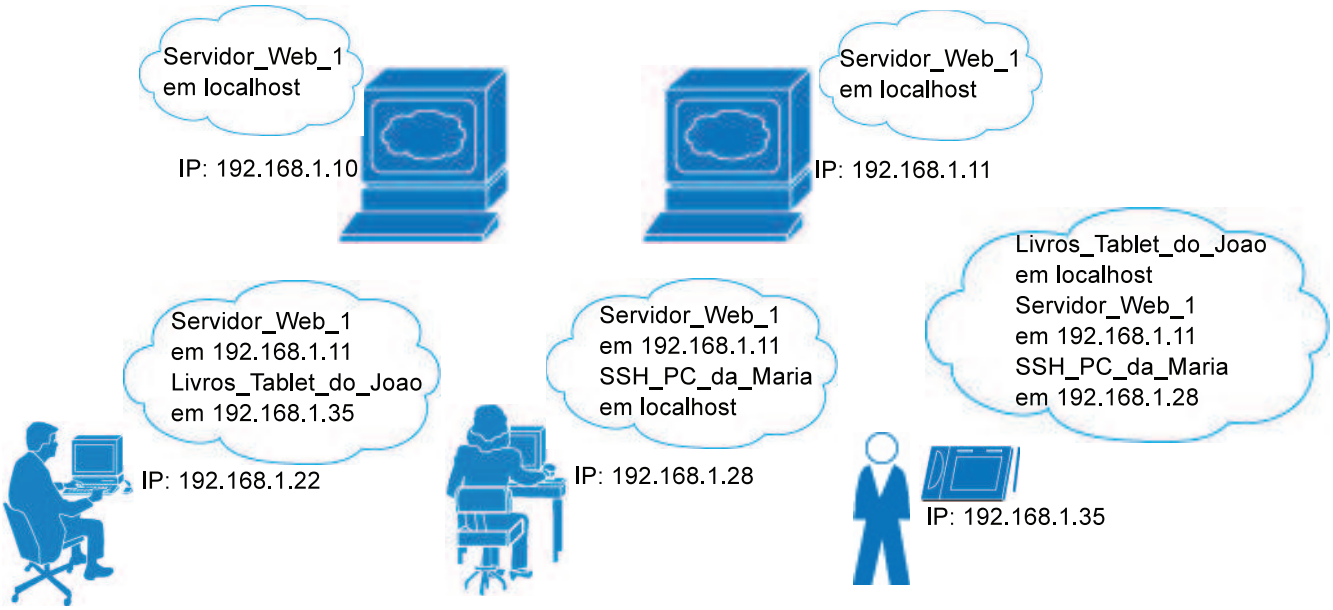

Figura 15: Rede com dois provedores redundantes para um serviço.

tempo de expiração que é pré-fixado pelo primeiro provedor que ingressa na rede.

A Figura 15 mostra dois provedores redundantes para o serviço Servidor_Web_1 Um cliente que requisita este serviço recebe a chave de grupo de um dos provedores. Se um deles deixar a rede, o Servidor_Web_1 continuará disponível, sendo atualizado somente o endereço IP em que este se encontra. O procedimento utilizado para este fim é descrito na seção *Obtenção de informações dos serviços*.

4.1.5.1 Adição de provedores redundantes

Conforme foi citado anteriormente, o proprietário de um serviço (por exemplo, o administrador da rede), atribui um identificador e uma senha para o mesmo e distribui estas credenciais aos clientes (usuários e/ou dispositivos) que poderão obter posteriormente informações sobre o serviço.

Quando é desejada a autenticação individual do cliente e do provedor no processo de descoberta, o proprietário deve distribuir também a sua chave pública e obter as chaves públicas dos clientes.

Um proprietário pode, a qualquer momento, ceder a um ou mais dispositivos o direito de atuar como provedor redundante para um determinado serviço. Neste caso, a informação de que existe redundância é armazenada em todos os provedores daquele serviço (ex. em um arquivo de configuração do protocolo Sec-SD).

Para os clientes, a existência ou não de redundância para os serviços é indiferente no que diz repeito as suas ações. Porém, no caso de ser desejada a autenticação do cliente e do provedor com criptografia assimétrica, todos os provedores redundantes devem receber as chaves públicas dos clientes e vice-versa.

4.1.5.2 Aquisição da chave de grupo em modo de redundância

No momento em que um provedor que atua em modo redundante entra na rede ele deve enviar uma requisição de autenticação para obtenção da chave de grupo, visto que a mesma já pode ter sido gerada por outro provedor. Se já existirem outros provedores na rede, estes deverão responder a requisição do mesmo modo que um provedor único na rede responderia para uma entidade que atua em modo cliente. Nas mensagens, existe um campo que é utilizado para informar se o dispositivo opera em *modo cliente* ou *modo de redundância*, o *OPCODE*, que é tratado na seção *Estrutura das mensagens do protocolo Sec-SD*.

Se for recebida mais de uma resposta, o requisitante deve escolher um provedor para se comunicar considerando o menor time-stamp e descartar silenciosamente as demais repostas. Por outro lado, se nenhuma resposta for recebida, ele deve enviar novamente a requisição. Se ainda assim não obtiver resposta, ele deve gerar a chave de grupo e aguardar por requisições de clientes e de outros provedores. O procedimento para análise das requisições de autenticação na existência de provedores redundantes é detalhado a seguir.

4.1.5.3 Análise das requisições de autenticação em modo de redundância

Com o objetivo de evitar um número excessivo de mensagens de descoberta em ambientes com grande número de nodos e/ou de serviços, os provedores que atuam em redundância aplicam uma probabilidade para processar as requisições, diferindo do comportamento de um provedor que é único.

Essa probabilidade é calculada em função da quantidade de serviços oferecidos em modo de redundância. Desta forma, um provedor que não atua em modo de redundância para nenhum serviço processará (e possivelmente responderá) todas as requisições, enquanto que um provedor que é redundante para um ou mais serviços processará um número menor de requisições, de acordo com a fórmula:

$$P_r = S/(S + C_r)$$

Onde S é o número de serviços disponíveis via Sec-SD e C_r é o coeficiente de redundância, que assume um valor inteiro igual a 1 ou maior, sendo único para cada máquina da rede e correspondendo ao número de serviços que cada provedor disponibiliza em modo de redundância. Para um provedor não redundante, C_r tem valor zero.

Por exemplo, para um provedor que atue em redundância e disponibilize 10 serviços, sendo 3 deles providos em redundância, ou seja, $C_r = 3$, a probabilidade de processamento de uma requisição P_r será de 0.77. Se na mesma máquina C_r fosse elevado para 7, P_r assumiria o valor de 0.59.

Com o objetivo de evitar que um cliente de um serviço que é oferecido em modo de re-

dundância fique sem resposta quando existe um número reduzido de provedores na rede (ex. um provedor somente), os provedores redundantes armazenam as requisições recebidas por um determinado período de tempo, que deve ser superior a duas vezes o valor do *timeout*, considerando-se que o cliente envia duas requisições antes de assumir que um provedor está indisponível. Se ele deixa de processar uma requisição e pouco tempo depois volta a recebê-la, ele processa imediatamente a requisição, ou seja, a probabilidade P_r assume valor um. Da mesma forma, um cliente que solicita um serviço que é oferecido com exclusividade por um provedor que é redundante para diversos outros serviços não ficará sem resposta, visto que ele envia duas vezes uma requisição antes de assumir que um serviço não está disponível.

Quando um provedor redundante envia a chave a um cliente ele envia o tempo pelo qual a chave ainda será válida e determina também o momento em que o cliente deverá requisitar a nova chave, tal como ocorre com os dispositivos que atuam em modo provedor.

Nos casos em que a chave de grupo é enviada a um provedor redundante, é enviada uma prioridade para este gerar a nova chave de grupo quando se aproxima o momento de expiração da chave atual. Este procedimento encontra-se detalhado na seção *Renovação da chave de grupo*.

4.2 Estrutura das mensagens do protocolo Sec-SD

A sequência de mensagens trocadas na interação entre as partes envolvidas para obtenção da chave de grupo não difere pelo fato de a descoberta ser realizada por um dispositivo que é cliente ou por um provedor redundante, havendo apenas variações nos conteúdos de alguns campos. Estas mensagens são explicadas abaixo.

Um cliente que possui privilégios para acessar um determinado serviço ingressa na rede e busca pelo mesmo enviando requisições multicast com o objetivo de obter a *chave de grupo*. A requisição enviada pelo cliente contém um hash do identificador do serviço concatenado com a senha do mesmo e com um time-stamp, que é aplicado com o objetivo de prevenir ataques causados pela repetição de mensagens capturadas de forma indevida por entidades mal-intencionadas.

Para possibilitar a verificação da mensagem por parte do recipiente, o time-stamp também é enviado em claro na mensagem. A requisição, assim como as demais mensagens, contém ainda um byte identificador do protocolo Sec-SD, um *OPCODE* para designar a mensagem como sendo uma *requisição de autenticação* e um campo *PRIO*, que será utilizado mais tarde pelos provedores redundantes no processo de renovação de chave de grupo.

Quando recebe uma requisição, o provedor de serviços verifica a mesma gerando um hash do identificador de cada um dos seus serviços concatenado com a respectiva senha e com o time-stamp recebido e compara com com o hash extraído da requisição. Se para algum dos seus serviços os dados correspondem, ele deve responder ao potencial cliente. A resposta enviada pelo provedor consiste do hash do identificador do serviço concatenado com a senha do mesmo

e com um novo time-stamp. Os dois time-stamps também são enviados em claro na mensagem.

O cliente verifica a resposta recebida de forma similar aquela feita pelo provedor e solicita então a chave de grupo. Novamente, o provedor verifica a requisição e se a mesma estiver conforme o esperado, ele envia a chave de grupo. A chave de grupo é valida por um tempo pré-determinado, como por exemplo uma hora a contar do momento em que ela foi gerada pelo provedor. O tempo faltante para a expiração em segundos, o tempo para renovação da chave e a prioridade para o processo de renovação da chave para provedores redundantes também são enviados com a chave.

Os campos presentes nas mensagens e a descrição detalhada das mesmas constam abaixo.

4.2.1 Campos das mensagens

Nesta seção são detalhados os campos das mensagens trocadas entre clientes e provedores de serviços durante o processo de descoberta.

OPCODE Este campo identifica o tipo de mensagem enviada e se é usada criptografia assimétrica em determinados campos da mesma, sendo utilizado também para informar se o dispositivo opera em *modo cliente* ou *modo de redundância*.

No *modo cliente*, a chave criptográfica recebida ao final da negociação com o provedor é utilizada para acesso ao serviço, enquanto que o *modo de redundância* é destinado aos provedores de serviço, que após a obtenção da chave passam também a responder requisições e enviar a chave a outros clientes.

Ao analisar este campo em uma requisição, um provedor que não atua em redundância pode, por exemplo, descartar uma mensagem que contenha um *OPCODE* destinado a provedores redundantes ou identificar o recebimento de uma solicitação de renovação de chave de grupo, assunto que é tratado nas seções a seguir, assim como o *modo de redundância*.

Na Tabela 2 são mostrados os possíveis valores para este campo nas mensagens enviadas quando um dispositivo opera em *modo cliente/provedor*.

Os valores para o *OPCODE* quando um provedor opera em *modo de redundância* são mostrados na tabela 3. Neste ponto, é importante salientar que sequência de mensagens trocadas entre cliente e provedor é muito similar para ambos os modos de operação.

ID Este campo identifica a mensagem como sendo pertencente ao protocolo Sec-SD e será analisado pelo cliente e pelo provedor de serviços antes da verificação das demais informações. Se o ID recebido contiver um valor inválido, a mensagem é descartada silenciosamente pela entidade que a recebeu.

PRIO Indica a prioridade do dispositivo para o processo de renovação da chave de grupo. Quando um dispositivo ingressa na rede, ele busca por serviços enviando requisições com

Tabela 2: Mensagens para os modos cliente e provedor.

Tipo de Mensagem	OPCODE	Significado
Autenticação	**1**	Requisição de autenticação
	2	Resposta de autenticação
	3	Requisição de autenticação para renovação de chave de grupo
	4	Resposta de autenticação para renovação de chave de grupo
Negociação	**11**	Solicitação de chave de grupo (autenticação para o grupo)
	12	Envio de chave de grupo (autenticação para o grupo)
	13	Solicitação de renovação de chave de grupo (autenticação para o grupo)
	14	Envio de chave de grupo renovada (autenticação para o grupo)
	21	Solicitação de chave de grupo (autenticação mútua com assinatura digital)
	22	Envio de chave de grupo (autenticação mútua com assinatura digital)
	23	Solicitação de renovação de chave de grupo (autenticação mútua com assinatura digital)
	24	Envio de chave grupo renovada (autenticação mútua com assinatura digital)
Descoberta	**31**	Descoberta (pedido de informações sobre um serviço - autenticação para o grupo)
	32	Resposta de descoberta (envio de informações sobre um serviço - autenticação para o grupo)
	41	Descoberta (pedido de informações sobre um serviço - autenticação mútua)
	42	Resposta de descoberta (envio de informações sobre um serviço - autenticação mútua)

Tabela 3: Mensagens para o modo de redundância.

Tipo de Mensagem	OPCODE	Significado
Autenticação	**131**	Requisição de autenticação
	132	Resposta de autenticação
	133	Requisição de autenticação para renovação de chave de grupo
	134	Resposta de autenticação para renovação de chave de grupo
Negociação	**141**	Solicitação de chave de grupo (autenticação para o grupo)
	142	Envio de chave de grupo (autenticação para o grupo)
	143	Solicitação de eleição para renovação de chave de grupo (envio de prioridade)
	144	Solicitação de renovação de chave de grupo (autenticação para o grupo)
	145	Envio de chave de grupo renovada (autenticação para o grupo)
	146	Solicitação de chave de grupo (autenticação mútual)
	147	Envio de chave de grupo (autenticação mútua)
	148	Solicitação de renovação de chave de grupo (autenticação mútua)
	149	Envio de chave de grupo renovada (autenticação mútua)
Descoberta	**151**	Descoberta (pedido de informações sobre um serviço - autenticação para o grupo)
	152	Resposta de descoberta (envio de informações sobre um serviço - autenticação para o grupo)
	153	Descoberta (pedido de informações sobre um serviço - autenticação mútua)
	154	Resposta de descoberta (envio de informações sobre um serviço - autenticação mútua)

o valor zero neste campo. O provedor de serviços sempre possui prioridade 1 e quando ele envia a chave de grupo a um cliente, o campo *PRIO* contém este valor. Os clientes sempre enviam o valor zero neste campo. Quando existem dispositivos atuando em modo de redundância na rede, o valor do campo *PRIO* indica a prioridade do provedor para gerar uma nova chave de grupo quando a chave atual está próxima de expirar. O processo para renovação de chave em modo de redundância é explicado na seção *Renovação da chave de grupo*.

HASH Este campo contém o hash obtido a partir das credenciais para acesso ao serviço (identificador do serviço e senha) e juntamente com os time-stamps T_{s1}, T_{s2} ou T_{s1} e T_{s2}, dependendo da mensagem. Estes time-stamps são aplicados com o objetivo tornar as mensagens dinâmicas e desta forma, prevenir ataques.

TS Contém o(s) time-stamp(s) a partir do(s) qual(is) foi obtido o hash descrito acima. A função deste campo é possibilitar a verificação da mensagem por parte do recipiente.

FPR Contém o *fingerprint* do cliente ou do provedor. Utilizado na *Solicitação de chave de grupo* nos casos em que é feita a autenticação mútua.

TVAL É o tempo em segundos pelo qual a chave de grupo que é enviada ainda será válida, calculado considerando o tempo de validade estipulado pelo provedor que gerou a chave.

TREN Indica o tempo em segundos em que um cliente que recebeu a chave de grupo deve solicitar a renovação da mesma junto ao provedor. Conforme foi explicado anteriormente, *TREN* é aplicado com o intuito de evitar que todos os clientes peçam uma nova chave no momento determinado por *TVAL*. Não é assumido que os relógios das máquinas presentes na rede estejam sincronizados, sendo este tempo para renovação relativo.

KEY Contém a chave de grupo criptografada.

INFO Contém dados específicos sobre os serviços, tais como endereço IP e porta, podendo conter também informações adicionais que são fornecidas pelos proprietários. Este campo é utilizado nas mensagens de descoberta que são enviadas quando o cliente já possui a chave de grupo. Uma explicação detalhada sobre o uso do *INFO* consta na seção *Obtenção de informações dos serviços*.

4.3 Protocolo para obtenção da chave de grupo

A sequência de mensagens trocadas entre as partes no início do processo de descoberta de serviços é mostrado na Figura 16. O conteúdo de cada uma das mensagens enviadas no processo de *autenticação para o grupo* é descrito abaixo.

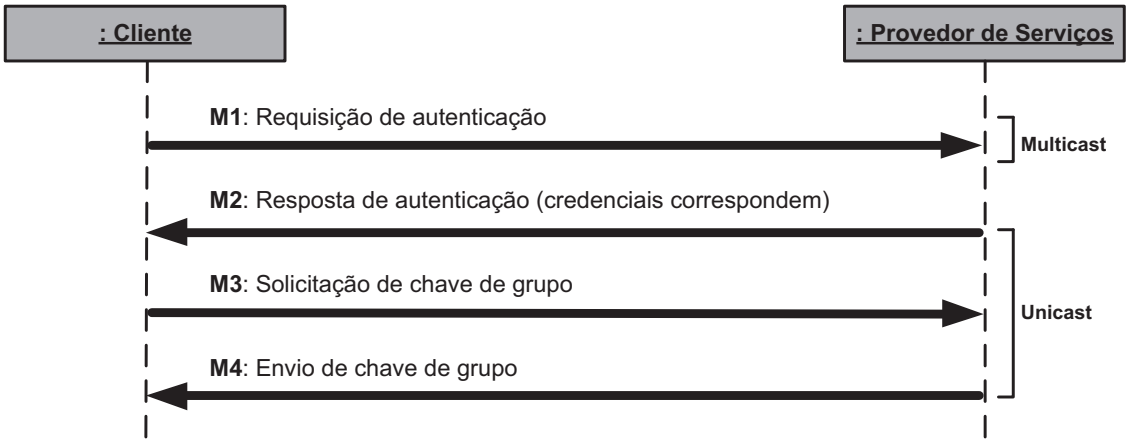

Figura 16: Diagrama de mensagens do Sec-SD para obtenção da chave de grupo.

1. **Requisição de autenticação** O cliente envia uma requisição em multicast contendo um identificador do protocolo Sec-SD, um *OPCODE*, um *ID*, um campo *PRIO* com valor zero, o campo *TS* com um time-stamp T_{s1} e o campo *HASH* com o hash obtido a partir do identificador do serviço concatenado com a senha do mesmo e com T_{s1}. *OPCODE* e *ID* são enviados em todas as mensagens, sendo por este motivo omitidos nas explicações a seguir.

2. **Resposta de autenticação** O provedor verifica o *OPCODE* da mensagem recebida e quando se trata de uma requisição por um serviço, ele extrai o hash e T_{s1} da requisição recebida e gera o hash de cada um dos seus identificadores de serviços concatenado a respectiva senha e com T_{s1}, comparando o resultado com o hash extraído da mensagem recebida. Se a requisição corresponde a algum dos seus serviços, o provedor envia uma resposta em unicast contendo o campo *PRIO* com o valor 1 (valor para provedores), *TS* com T_{s1} e um novo time-stamp T_{s2} e o *HASH* contendo o hash do identificador do serviço concatenado com a senha do mesmo e com T_{s2}.

3. **Solicitação de chave de grupo** O cliente extrai T_{s1} e T_{s2} e verifica se o hash recebido corresponde a algum serviço requisitado por ele. Se não corresponder, ele descarta a mensagem. O cliente solicita a chave de grupo enviando uma mensagem em unicast contendo o valor zero no campo *PRIO*, *TS* contendo T_{s1} e T_{s2} e *HASH* com o hash do identificador do serviço concatenado com a sua senha, com T_{s1} e com T_{s2}.

   As mensagens *Requisição de autenticação*, *Resposta de autenticação* e *Solicitação de chave de grupo* possuem os mesmos campos, diferindo apenas no conteúdo destes. O formato destas mensagens é mostrado na Figura 17

4. **Envio de chave de grupo** O provedor verifica a requisição conforme descrito em 2, descartando a mensagem se qualquer um dos campos contiver um dado inconsistente. Se a solicitação estiver correta, o provedor envia uma mensagem contendo valor 1 no

Figura 17: Campos das mensagens *Requisição* e *Resposta de autenticação* e *Solicitação de chave de grupo*.

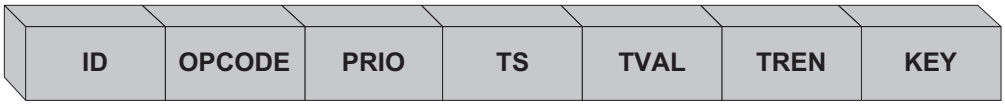

Figura 18: Mensagem *Envio de chave de grupo*.

campo *PRIO*, *TS* contendo T_{s2} e T_{s2}, o tempo restante para a expiração da chave em segundos no campo *TVAL*, o tempo em segundos para renovação da chave no campo *TREN* quando se tratar do envio de chave para um cliente e valor zero quando a chave for enviada para um provedor, e a chave de grupo e T_{s2} e criptografados com a senha do serviço no campo *KEY*. A figura 18 ilustra os campos presentes nesta mensagem.

4.3.1 Autenticação do cliente e do provedor de serviços

A negociação descrita acima permite que o cliente e o provedor de serviços verifiquem se o outro lado realmente pertence ao grupo que possui as credenciais para acessar um determinado serviço.

Isto é conseguido através do envio do hash do identificador do serviço com a senha do mesmo e com um time-stamp diferente em cada mensagem trocada pelas partes, com o objetivo de prevenir ataques causados pela repetição de mensagens por entidades mal-intencionadas. Além disso, se em qualquer etapa da negociação para obtenção da chave de grupo alguma das partes receber uma mensagem com informações inválidas em qualquer um dos campos, a mensagem é silenciosamente descartada.

Ainda que o nível de segurança obtido com estes procedimentos seja bastante alto, os clientes e provedores não são identificados de forma individual, o que é desejável em muitos casos. Por exemplo, em ambientes como pequenos escritórios ou redes domésticas, a negociação descrita anteriormente tende a ser suficiente. Por outro lado, em ambientes corporativos, é imprescindível que a autenticação seja feita de forma individual, permitindo identificar qual usuário (ou máquina) teve acesso a quais recursos.

Além disso, alguns ambientes são mais propícios a ataques provenientes do próprio grupo que tem acesso ao serviço (ataques internos). Neste caso, um membro do grupo que é cliente poderia responder as requisições enviando uma chave de grupo inválida ou até mesmo disponibilizando um falso serviço, uma vez que ele possui o identificador do serviço e a senha.

Para as situações descritas acima, o Sec-SD conta com um segundo nível de segurança,

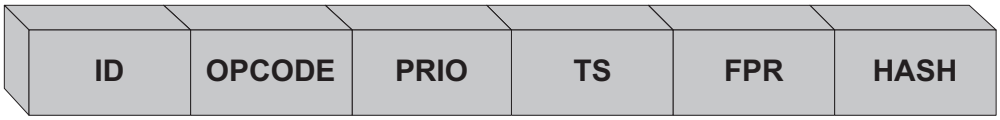

Figura 19: Mensagem *Solicitação de chave de grupo na autenticação mútua*.

onde o cliente e o provedor de serviço são identificados de forma individual através do uso de criptografia assimétrica.

Neste caso, o procedimento realizado difere do anterior pelo fato de que o provedor de serviço repassa aos clientes a sua chave pública e obtém as deles (além de informar aos clientes o identificador e senha do serviço, como no processo anterior). Para ser autenticado, o cliente assina com a sua chave privada a solicitação da chave de grupo. Da mesma forma, a chave de grupo é assinada pelo provedor de serviços e criptografada com a chave pública do cliente antes de ser enviada pela rede. A sequência de mensagens é descrita abaixo.

1. **Requisição de autenticação** e **Resposta de autenticação** Possuem a mesma estrutura das mensagens enviadas na *autenticação para o grupo*, que foram descritas anteriormente.

2. **Solicitação de chave de grupo** O cliente extrai T_{s1} e T_{s2} e verifica se o hash recebido corresponde a algum serviço requisitado por ele. Se não corresponder, ele descarta a mensagem. O cliente solicita a chave de grupo enviando em unicast uma mensagem com o valor zero no campo *PRIO*, *TS* contendo T_{s1} e T_{s2}, *FPR* com o seu *fingerprint* e as seguintes informações assinadas com a sua chave privada no campo *HASH*: o hash do identificador do serviço concatenado com a senha do mesmo e com T_{s1} e T_{s2}. O formato desta mensagem é mostrado na Figura 19.

3. **Envio de chave de grupo** O provedor de serviços extrai o *fingerprint* para identificar qual cliente enviou a mensagem e depois verifica o hash e os time-stamps recebidos. Por fim, o provedor envia uma mensagem contendo o valor 1 no campo *PRIO* quando se tratar do envio de chave para cliente e valores maiores do que 1 quando a chave for enviada para um provedor redundante, *TS* contendo T_{s2} e T_{s2}, o tempo restante antes da expiração da chave em segundos em *TVAL*, o tempo em segundos para renovação da chave no campo *TREN* quando se tratar do envio de chave para um cliente e valor zero quando a chave for enviada para um provedor, o seu *fingerprint* em *FPR* e as seguintes informações assinadas com a sua chave privada e criptografadas com a chave pública do cliente no campo *KEY*: a chave de grupo e T_{s2}. A figura 20 ilustra os campos desta mensagem.

Com a realização dos procedimentos descritos, é possível para o provedor de serviços autenticar o cliente de forma individual e não somente como um cliente pertencente ao grupo que tem acesso a um determinado serviço. De modo similar, os clientes podem ter certeza de que as informações referentes a um dado serviço são fornecidas por um provedor legítimo. Assim, no

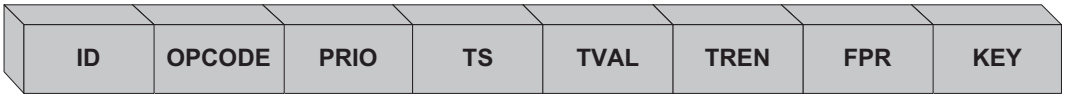

Figura 20: Mensagem *Envio de chave de grupo* na autenticação mútua.

momento da criação de um serviço o proprietário pode definir se utilizará a autenticação para o grupo que possui acesso ao serviço ou a autenticação mútua dos clientes e provedores, que caracterizam respectivamente os *níveis de segurança 1 e 2*.

Nos ambientes onde não há necessidade de autenticar cada máquina individualmente e/ou não é desejável para os provedores fornecer a chave pública (e obter a mesma) para cada dispositivo que tem acesso a um determinado serviço pode ser utilizado o nível 1 de segurança, enquanto que para os ambientes onde é indispensável identificar cada usuário ou máquina que obtém informações sobre um serviço e acessa o mesmo, como é o caso dos ambientes corporativos, é recomendável o uso do segundo nível de segurança.

No que diz respeito a mensagens de erro, o Sec-SD não retorna nenhum tipo de notificação quando ocorre um erro em qualquer uma das etapas. Este procedimento é realizado por motivos de segurança. Se em qualquer uma das mensagens for recebido um conteúdo inválido em qualquer campo, a entidade que recebeu a mensagem a descarta silenciosamente.

Por exemplo, se um cliente envia uma *Requisição de autenticação*, obtém a respectiva resposta, envia a *Solicitação de chave de grupo*, mas não obtém resposta, ele deve reiniciar o processo de autenticação.

4.4 Renovação da chave de grupo

Conforme foi citado anteriormente, a chave de grupo é válida por um tempo estabelecido pelo provedor de serviços no momento em que a mesma é gerada. Abaixo, são explicados os processos de renovação de chave para os modos *cliente* e de *redundância*.

4.4.1 Renovação da chave em modo cliente

Quando está operando sem redundância, o provedor deve gerar uma nova chave quando ainda resta mais de 5% do tempo para a expiração da chave atual e então aguardar por requisições.

Conforme foi explicado nas seções anteriores, no momento em que um dispositivo ingressa na rede, ele busca por serviços enviando requisições com o valor zero no campo *PRIO*. O provedor de serviços sempre possui prioridade 1 e quando ele envia a chave de grupo a um cliente. No campo *TREN*, as requisições para obtenção da chave contém valor zero e as requisições para renovação da chave contém o valor de *TREN* recebido do provedor junto com a chave. No que diz respeito ao campo *OPCODE*, este possui valores específicos para a operação de renovação

Figura 21: Mensagem *Requisição de autenticação para renovação de chave de grupo em modo cliente*.

da chave de grupo.

Todos os clientes que obtiveram uma chave de grupo devem enviar requisições multicast para o provedor quando a validade da chave estiver próxima do fim com o objetivo de obter uma nova chave. As requisições são enviadas em multicast por que para os serviços oferecidos em redundância o provedor que forneceu a chave para um determinado cliente pode não estar mais disponível e dessa forma, a requisição multicast pode ser respondida por outro provedor.

O momento preciso em que a requisição deve ser enviada é determinado pelo valor enviado pelo provedor ao cliente no campo *TREN*, junto com a chave. Este tempo deve ser correspondente a 99% do tempo de validade da chave, no máximo. Por exemplo, para uma chave válida por 3600 segundos (ou uma hora), o provedor pode gerar uma nova chave quando faltam 360 segundos ou menos para a expiração da chave atual. Os clientes verificam o valor recebido no campo *TREN* e enviam suas requisições quando faltam menos de 180 segundos para a expiração, até que reste no mínimo 36 segundos (ou 1%) do tempo de validade da chave atual.

Este esquema é empregado com o objetivo de evitar tráfego excessivo durante a renovação de chave de grupo em ambientes com muitos nodos, visando tornar o Sec-SD escalável.

Para renovar a chave de grupo, os cliente enviam requisições muito similares àquelas enviadas no momento em que eles ingressam na rede. Neste caso, o campo *TS* contendo o valor recebido em *TREN* do provedor junto com a chave. Novamente, o cliente deve gerar hashes da combinação das credenciais dos serviços com um time-stamp. Assim, os campos restantes são idênticos ao da mensagem *Requisição de autenticação*, conforme mostra a Figura 21

Todas as requisições são processadas imediatamente pelos provedores e respondidas obedecendo ao mesmo critério da mensagem *Resposta de autenticação*. O formato das mensagens *Requisição e Resposta de autenticação para renovação de chave de grupo* são os mesmos. Estas mensagens descritas acima são idênticas para os dois níveis de autenticação usados no Sec-SD.

O cliente verifica a resposta e solicita a nova chave de grupo com a mensagem *Solicitação de renovação de chave de grupo*, que possui o mesmo formato da mensagem *Solicitação de chave de grupo* utilizadas nos níveis de segurança 1 e 2. Se for recebida mais de uma resposta ele considera a de menor time-stamp e descarta as demais.

O provedor, por sua vez, verifica a mensagem e responde com a *Renovação de chave de grupo*, que contém um novo *TREN*. Os demais campos são iguais ao da mensagem *Envio da chave de grupo*, respeitando o nível de segurança utilizado no serviço.

As etapas explicadas acima permitem que um cliente possa obter uma nova chave de grupo

sem interromper sua comunicação com o serviço, o que é conseguido através da solicitação de uma nova chave no tempo programado pelo provedor que lhe forneceu a chave original.

4.4.2 Renovação da chave em modo de redundância

Quando está operando em modo de redundância, os provedores devem iniciar a eleição para definir quem deve gerar uma nova chave quando resta 10% do tempo para a expiração da chave atual e então aguardar por requisições.

Conforme foi explicado nas seções anteriores, no momento em que um dispositivo ingressa na rede, ele busca por serviços enviando requisições com o valor zero no campo *PRIO*. O primeiro provedor de serviços que ingressa na rede (e gera a chave de grupo) sempre possui prioridade 1. Quando ele envia a chave de grupo a um provedor redundante, a prioridade deste para gerar a nova chave é acrescida de um. Deste modo, o primeiro provedor redundante a receber a chave terá prioridade 2 para gerar a nova chave de grupo, o segundo terá prioridade 3 e assim por diante.

Os valores deste campo estão associados a ordem de ingresso na rede, de modo que os provedores com prioridade maior (isto é, com valor superior) são aqueles que entraram mais recentemente, sendo mais suscetíveis a estarem ativos no momento da eleição. Por este motivo, o nodo com prioridade de valor mais alto vence a eleição e gera a nova chave.

Quando um provedor que atua em redundância fornece a chave de grupo para um cliente, envia também o momento em que ele deve solicitar uma nova chave, do mesmo modo que ocorre quando há somente um dispositivo que exerce o papel de provedor na rede (isto é, sem redundância). É importante destacar que a diferenciação entre um cliente que requisita uma nova chave e um provedor redundante que requisita a eleição para renovação de chave é feita através da verificação campo *OPCODE*, seguida da verificação dos campos *TS* e/ou *PRIO*, de acordo com cada caso, *HASH* e *TS*.

Um provedor redundante que deseja iniciar o processo de eleição para renovar a chave envia a mensagem *Solicitação de eleição para renovação de chave de grupo* que contém no campo *PRIO* a sua prioridade para gerar da nova chave de grupo. Os provedores redundantes que estão na rede verificam estes campos e se tiverem uma prioridade maior (isto é, com valor superior), respondem com a mesma, permanecendo em silêncio caso contrário. O momento em que os provedores devem enviar suas *Solicitações de eleição* é quando resta 10% do tempo de validade da chave. Os provedores respondem com mensagens no mesmo formato, contendo as suas prioridades no campo *PRIO*.

A Figura 22 mostra a sequência de mensagens enviadas para renovação da chave de grupo. Por exemplo, podemos considerar que existem na rede os provedores A, B, C e D, respectivamente com as prioridades 1, 2, 3, e 4. O provedor B inicia a eleição enviando sua prioridade 2 e sua mensagem é processada pelos demais conforme segue: A possui prioridade 1 (sendo o mais antigo na rede) e por isso permanece em silêncio. C responde com *PRIO = 3*, fazendo

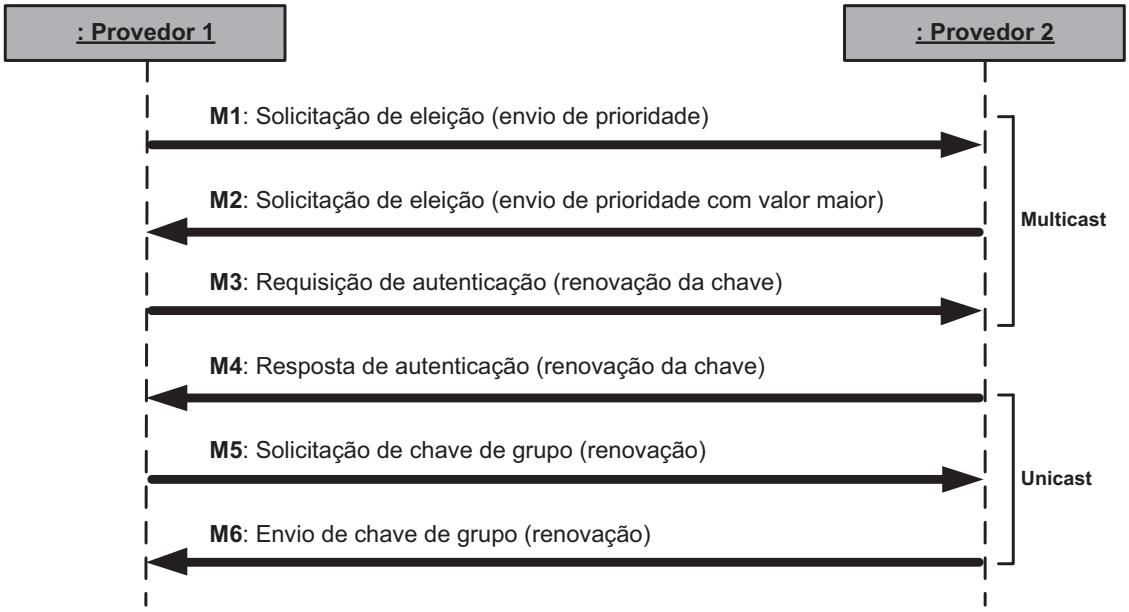

Figura 22: Diagrama de mensagens para renovação da chave de grupo em modo de redundância.

com que A e B permaneçam em silêncio e D responda com *PRIO = 4*. Como D possui a maior prioridade, nenhum outro provedor redundante responderá e ele deverá gerar a nova chave. Por outro lado, se D tivesse iniciado a eleição, todos os outros permaneceriam em silêncio.

Um provedor que envia uma *Solicitação de eleição* e não recebe respostas antes que seja atingido o *timeout* (da ordem de segundos), deve enviar uma nova mensagem. Se ainda assim não receber respostas, ele assume que é o provedor com maior prioridade (que entrou mais recententemente na rede) e gera a nova chave de grupo.

Quando resta 7% do tempo para a chave expirar os demais provedores podem requisitar a nova chave de grupo, com mensagens muito similares as que foram enviadas quando eles ingressaram na rede, contendo *PRIO = 0*, *HASH* com os hash obtido das credenciais e de um time-stamp, etc. O provedor processa as requisições obedecendo aos mesmos critérios da mensagem *Requisição de autenticação* já descrita e responde com a mensagem *Resposta de autenticação para renovação de chave de grupo*. O cliente envia então a *Solicitação de renovação de chave de grupo* e, finalmente, recebe a chave na mensagem *Renovação de chave de grupo*.

Devido ao fato de o número de provedores redundantes em uma rede ser usualmente menor do que a quantidade de clientes, não existe nenhuma restrição quanto ao momento em que estes devem enviar suas *(Requisições de autenticação para renovação de chave de grupo)*. No que diz respeito aos clientes, estes deverão enviar suas requisições quando resta 5% a 1% do tempo de validade da chave ou menos, de acordo com o valor recebido em *TREN* junto com a chave.

Para as redes onde todos os dispositivos atuam como provedores redundantes (como por exemplo, em aplicações P2P), o provedor redundante recebe juntamente com a chave de grupo a sua prioridade para a renovação da mesma e ainda, o tempo para requisitar a nova chave

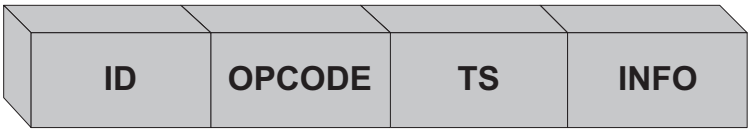

Figura 23: Mensagem *Descoberta* para o nível de segurança 1.

Figura 24: Mensagem *Descoberta* para o nível de segurança 2.

no campo *TREN*, caso ele não seja o vencedor da eleição para renovação da chave. Dessa forma, é possível evitar que muitos nodos enviem requisições ao mesmo tempo. A informação de que todos os provedores são redundantes pode constar, por exemplo, em um arquivo de configuração.

4.5 Obtenção de informações dos serviços

De posse da chave de grupo, um cliente autenticado com uso dos níveis de segurança 1 ou 2 já esta ciente sobre a disponibilidade de um serviço e pode então obter informações relacionadas a este, tais como endereço IP, porta e outros dados fornecidos pelo provedor. As mensagens enviadas nesta etapa são descritas a seguir.

1. **Descoberta** O cliente envia uma requisição em multicast contendo os campos *OPCODE* e *ID* (identificador do protocolo Sec-SD). Além destes, são enviados os campos *TS* com um time-stamp T_{s3} e *INFO*, onde deve constar o usuário (ex. o nome da máquina). *TS* e *INFO* são criptografados com a chave de grupo para clientes autenticados com uso do primeiro nível de segurança, conforme mostra a Figura 23.

   Os clientes autenticados com o segundo nível de segurança devem enviar o seu *fingerprint* no campo *FPR*, localizado após o *OPCODE*, assinar *TS* e *INFO* com a sua chave privada e em seguida criptografá-los com a chave de grupo, como consta na Figura 24.

2. **Resposta de descoberta** O provedor verifica o *OPCODE* da mensagem recebida, e tenta descriptografar os campos *TS* e *INFO*. Se a requisição corresponde a algum dos seus serviços, o provedor envia uma resposta em unicast contendo *ID*, *OPCODE* e *TS* com T_{s3} e um novo time-stamp T_{s4}. O campo *INFO* contém o endereço IP do provedor, a porta onde o serviço está disponível, o usuário (ex. o nome da máquina), podendo conter ainda dados adicionais sobre o serviço, sendo estes de livre escolha do proprietário.

   Novamente, *TS* e *INFO* são criptografados com a chave de grupo para clientes autenti-

cados com nível de segurança 1. No caso de as partes serem autenticadas com nível de segurança 2, estes campos devem ser assinados pelo provedor em seguida criptografados com a chave de grupo. Além disso, o *fingerprint* deve ser enviado.

Os campos presentes nas *Respostas de descoberta* são os mesmos das mensagens *Descoberta*.

A mensagem *descoberta* deve ser enviada logo após o cliente se autenticar e receber a chave de grupo. Ao receber a resposta do provedor, ele estará apto a acessar o serviço em questão. Se não receber nenhuma resposta em alguns segundos (de acordo com o *timeout* especificado pelo administrador da rede ou na implementação), o cliente envia novamente a mensagem. Se ainda assim não receber respostas, ele assume que o provedor deixou a rede. Neste caso, ele poderá enviar solicitações subsequentes desse mesmo modo durante o tempo de validade da chave, respeitando um intervalo de tempo também configurado pelo administrador da rede. No caso dos provedores redundantes, estes podem enviar mensagens de descoberta com o objetivo de ter um controle sobre a atividade dos demais provedores do mesmo serviço.

Quando o cliente recebe resposta, ele deve continuar enviando mensagens *descoberta* periodicamente para verificar a disponibilidade do provedor, uma vez que este não se anúncia e não notifica sua desativação. O intervalo entre as mensagens deve ser menor que $TVAL/10$ e pode ser configurado para cada serviço, possibilitando que o usuário possa tomar conhecimento de forma mais rápida sobre a desativação de serviços mais críticos.

Para os serviços oferecidos por provedores redundantes, o processamento das mensagens *descoberta* é feito obedecendo aos mesmos critério aplicado as *requisições de autenticação*, sendo aplicada a probabilidade $P_r = S/(S+C_r)$ no recebimento de uma mensagem na primeira vez e $P_r = 1$ na segunda vez. Para os provedores redundantes que enviam estas mensagens, o processamento das mesmas é feito com $P_r = 1$ (o modo de operação é informado no campo *opcode*.

Desse modo, um serviço disponibilizado com redundância só é reconhecido como indisponível quando realmente não existir nenhum provedor seu na rede. A cada solicitação enviada pelo cliente com o objetivo de verificar a disponibilidade do serviço, ao menos um provedor deve enviar resposta (a chave que criptografa a mensagem é válida junto a todos os provedores de um serviço), sendo que o cliente considera a resposta com menor time-stamp e descarta as demais.

O envio do nome de usuário acompanhado de um time-stamp nestas mensagens possibilita a criação de arquivos de *log* no lado dos clientes e dos provedores de serviço. Desta forma, é possível que o administrador da rede tenha conhecimento sobre o momento exato dos acessos as informações dos serviços realizados pelos clientes, bem como as atividades dos provedores.

5 METODOLOGIA E VALIDAÇÃO

Neste capítulo são mostrados os meios utilizados para validar o modelo Sec-SD e os resultados obtidos. São discutidos aspectos relacionados a confiabilidade e ao desempenho do sistema. É apresentada também uma avaliação da quantidade de mensagens do protocolo que trafegam pela rede e os resultados obtidos com a execução do protótipo desenvolvido.

5.1 Confiabilidade do modelo Sec-SD

Conforme foi descrito anteriormente, um dispositivo que executa o Sec-SD ingressa na rede e procura por cada um dos serviços para os quais ele possui credenciais enviando requisições em multicast que contem os hashes de tais credenciais com o objetivo de obter as chaves de grupo para poder então adquirir informações sobre os serviços.

Durante este processo não ocorrem falsos negativos, isto é, sempre que um usuário solicitar um serviço para o qual ele possui as credenciais, se houver um provedor na rede, este irá responder e dar continuidade a autenticação do cliente. Os falsos negativos não ocorrem por que tanto o dispositivo que envia a requisição quanto aqueles que a recebem possuem todas as informações necessárias (no caso, as credenciais do serviço) para verificar se o outro lado representa uma entidade válida. Assim, os provedores sempre autenticam os clientes que são válidos e os provedores válidos sempre tem suas respostas aceitas pelos clientes.

No que diz respeito aos falsos positivos, para uma entidade se autenticar junto a um provedor para um serviço do qual ela não possui as devidas credenciais, é necessário que a mesma gere um hash idêntico ao que seria produzido pela entrada do identificador do serviço, senha e o time-stamp. Devido a aplicação do time-stamp, as mensagens são dinâmicas, o que reduz a possibilidade de que ocorram falsos positivos. Além disso, para obter a chave de grupo, o intruso deve interpretar a resposta recebida do provedor e gerar um novo hash para ser inserido na solicitação de chave de grupo.

A chave de grupo é enviada criptografada com a senha do serviço ou com a chave pública do cliente, dependendo do nível de segurança empregado. De posse desta, o cliente pode enviar requisições de descoberta para obter informações relacionadas ao serviço. A chave é trocada periodicamente mediante um novo processo de autenticação, porém sem a interrupção do acesso as informações dos serviços, que só ocorre se a chave expirar sem que o cliente tenha se autenticado para renovar a mesma.

Com o uso destes mecanismos de segurança, os ataques decorrentes do envio de pacotes capturados indevidamente durante as interações entre clientes e provedores legítimos são minimizados de forma significativa. Além disso, a possibilidade de ataque interno (proveniente de um cliente que tem acesso ao serviço e tenta se passar por provedor) pode ser anulada com o uso de autenticação mútua através de assinaturas digitais nos ambientes onde esta funcionalidade for considerada conveniente.

5.2 Análise do tráfego gerado pelo Sec-SD

Nesta seção é apresentada uma avaliação do tráfego gerado pelo protocolo Sec-SD com base na quantidade de mensagens enviadas durante o processo de autenticação para obtenção da chave de grupo.

Com o intuito de verificar a adequação do Sec-SD para ambientes com grande quantidade de nodos e de serviços, foi avaliado o impacto do tráfego causado pelas mensagens de resposta de autenticação enviadas em unicast pelos provedores. Estas mensagens foram escolhidas pelo fato de serem enviadas no momento em que os dispositivos ingressam na rede e iniciam o processo para obtenção das chaves de grupo, não sendo aplicada nenhuma regra para conter o volume de respostas dos provedores de serviços que não atuam em redundância. As demais mensagens são enviadas em tempos pré-definidos pelos provedores (ex: renovação de chave) ou respeitando intervalos regulares (ex: descoberta) e desta forma, contribuem de forma menos significativa para o aumento do tráfego na rede.

Os cenários utilizados são descritos a seguir.

5.2.1 Descoberta de serviços providos de forma exclusiva e com redundância

Para verificar como as respostas de autenticação enviadas pelos provedores contribuem para o aumento do tráfego em uma rede, consideramos o seguinte ambiente:

Uma rede é composta por uma quantidade variável de nodos clientes. Cada nodo pode acessar somente os serviços para os quais ele possui credenciais. Neste cenário, existem dois tipos de provedores:

- Os provedores que oferecem com exclusividade todos os seus serviços, ou seja, não operam em redundância;

- Os provedores que são redundantes para pelo menos um de seus serviços, e que ao receberem uma requisição necessitam aplicar o cálculo para determinar a probabilidade de resposta P_r, que é dada por

  $$P_r = S/(S + C_r)$$

  onde C_r é o coeficiente de redundância, que assume um valor igual a 1 ou maior e é único para cada máquina da rede, correspondendo ao número de serviços que cada uma disponibiliza em modo de redundância.

Os clientes, por sua vez, acessam uma determinada quantidade de serviços providos de forma exclusiva e outra parcela de serviços oferecidos em redundância.

A escolha deste cenário ocorreu por que a existência de provedores redundantes contribui para a ocorrência do envio de respostas em unicast repetidas quando um cliente faz uma requisição, pois mesmo com a aplicação da fórmula acima para determinar se um dispositivo deve

processar (e possivelmente responder) ou não a uma requisição, podem ocorrer respostas repetidas.

Quando um dispositivo ingressa na rede, ele requisita cada um dos serviços para os quais ele possui credenciais enviando mensagens em multicast. Os serviços requisitados podem ou não estar disponíveis no momento em que o novo membro envia suas requisições. Sempre que um provedor que é exclusivo recebe uma requisição contendo as informações corretas ele irá responder. Para modelar este comportamento, utilizamos o coeficiente d, que compreende a disponibilidade de serviços providos de forma exclusiva na rede em um dado momento. No caso dos provedores redundantes, estes recebem as requisições e antes de processá-las, aplicam a fórmula para calcular P_r.

Assim, o tráfego de mensagens do tipo resposta de autenticação unicast em um determinado instante é dado por

$$M = N * (S_e * d + S_r * P_r)$$

onde N é a quantidade de nodos clientes que descobrem tanto serviços exclusivos quanto serviços redundantes na rede, S_e é a quantidade média de serviços providos de forma exclusiva e S_r é a quantidade média de serviços providos em redundância que cada nodo pode descobrir (ou seja, possui as credenciais). Os dispositivos denotados por N são máquinas que executam o Sec-SD e atuam ao menos como clientes, podendo ou não disponibilizar serviços, d compreende a disponibilidade média de serviços exclusivos na rede em um dado momento e P_r é a probabilidade de um provedor redundante processar uma requisição, tendo um valor médio para a rede nos cenários utilizados.

Para os serviços disponibilizados de modo exclusivo, P_r sempre assume valor 1 (sendo omitido no cálculo), enquanto que $0.1 <= d <= 1$, representa o percentual médio de serviços que estão disponíveis em um determinado momento. Para o conjunto de provedores redundantes, assume-se $d = 1$, ou seja, existe um pelo menos um provedor ativo, que processará as requisições conforme P_r determina. Assim, a contribuição dos dois tipos de provedores, únicos e redundantes, para o aumento do tráfego é denotada respectivamente por d e P_r.

Os cenários considerados são descritos abaixo.

Cenário 1: Neste caso, é considerado que cada um dos provedores ativos na rede disponibiliza em média 5 serviços, sendo 20% deles providos em modo de redundância. Assim, o cálculo de P_r fornece o valor que consta abaixo. Conforme ocorre nos demais cenários, cada dispositivo descobre a mesma quantidade de serviços exclusivos e redundantes.
$100 <= N <= 400$
$S_e = S_r = 30$
$d = 0.95$
$P_r = 0.83$

Cenário 2: Para este experimento foi considerado um ambiente onde cada provedor fornece em mé-

dia 10 serviços, sendo 40% deles oferecidos em modo de redundância.
$100 <= N <= 400$
$S_e = S_r = 60$
$d = 0.95$
$P_r = 0.71$

Cenário 3: Neste cenário, cada provedor fornece 20 serviços e 70% deles são disponibilizados em redundância.
$100 <= N <= 400$
$S_e = S_r = 90$
$d = 0.95$
$P_r = 0.58$

Estes valores foram escolhidos com o objetivo de representar redes de tamanho variável no que diz respeito a quantidade de nodos e de serviços disponibilizados e o impacto do uso do Sec-SD nestes ambientes, avaliando assim a adequação do mesmo às redes com grande número de máquinas, onde o tráfego de mensagens aumenta. Também por estes motivos, o valor utilizado para d é alto, representando disponibilidade média de 95% dos serviços requisitados pelos clientes.

Como exemplo de ambiente onde o coeficiente de redundância de serviços tende a ser alto, pode-se citar uma rede na qual as máquinas executam o sistema LP2P (*Local Peer-to-Peer Protocol)*. Uma explicação detalhada sobre o uso do modo de operação em redundância no LP2P é descrito na seção *Caso de uso: Descoberta de serviços em um sistema Peer-to-Peer projetado para ambientes LAN*.

Entre as mensagens do Sec-SD foram destacadas as respostas de autenticação para esta avaliação pelo fato de estas serem enviadas no momento em que os dispositivos ingressam na rede e iniciam o processo para obtenção as chaves de grupo. Assim, não é realizado nenhum procedimento para reduzir a quantidade de respostas dos provedores de serviços que não atuam em redundância. Estas mensagens tem tamanho de 58 bytes. As demais mensagens são enviadas em momentos pré-definidos pelos provedores ou em intervalos regulares, não causando impacto significativo no tráfego.

Considera-se que os provedores de serviços estão ativos conforme indicam as variáveis d e P_r no momento em que os clientes ingressam na rede. O gráfico correspondente ao tráfego de mensagens em função do número de dispositivos e de serviços na rede para os três cenários, que foi obtido a partir do uso das fórmulas apresentadas e dos valores acima, é mostrado na Figura 25.

A quantidade de mensagens que é mostrada no gráfico representa uma situação onde a entrada dos clientes ocorre em um intervalo de 30 minutos. Um comportamento como esse poderia

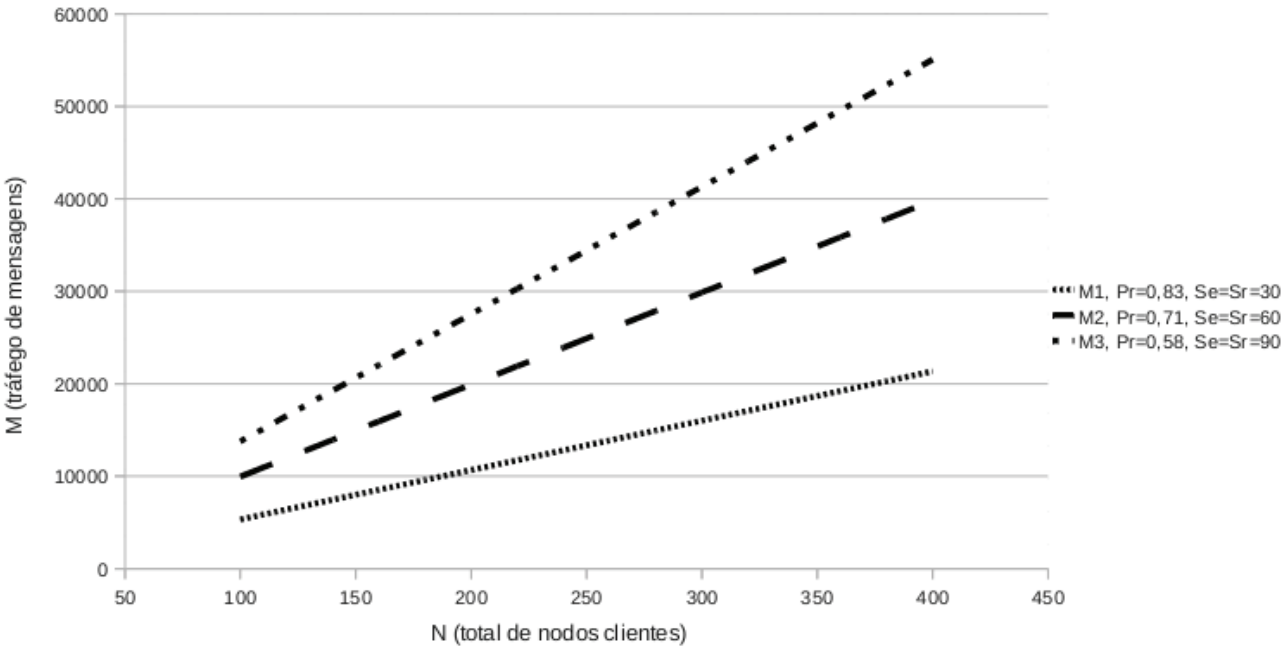

Figura 25: Tráfego de respostas de autenticação do Sec-SD.

ocorrer em uma empresa, no momento da chegada dos funcionários ou em uma universidade, durante a chegada de alunos com seus notebooks em uma sala de aula e/ou ligando as máquinas de um laboratório. Com a consideração deste intervalo para a análise, é possível obter uma estimativa do tráfego de pacotes Sec-SD por minuto na rede em situações onde o volume de mensagens enviadas tende a apresentar picos.

Observou que com o cenário 1 (gráfico M1), com 200 nodos ativos na rede (caso médio) descobrindo um total de 60 serviços ($S_e + S_r$) a média de respostas de autenticação que são enviadas em unicast é de 356 pacotes por minuto. Com 400 nodos ativos (pior caso), este número sobe para 712 pacotes a cada minuto. Neste caso, por existirem poucos serviços (1 por provedor) sendo oferecidos em modo de redundância, a probabilidade de resposta dos provedores é alta.

Já no cenário 2 (de acordo com o gráfico M2), com 200 nodos clientes na rede descobrindo 120 serviços no total, a média de mensagens do tipo respostas de autenticação é de 664 pacotes por minuto. Com 400 nodos ativos, são enviados 1328 pacotes a cada minuto. Neste cenário 40% dos serviços são oferecidos em redundância pelos provedores, implicando em uma probabilidade de resposta de 71%.

Por último, tem-se no cenário 3 uma rede onde os nodos descobrem um total de 180 serviços e os provedores atuam em redundância 70%. Neste caso, são enviadas em média 918 mensagens por minuto quando se tem 200 clientes ativos e 1836 quando existem 400 nodos ingressando na rede em um intervalo de 30 minutos e descobrindo serviços.

Considerando-se que a maior quantidade de tráfego gerado pelo Sec-SD nos casos descritos

é de 91KB por minuto $(1836 * 50)$ bytes , correspondendo ao cenário 3, em uma rede de 100 Mb/s, é possível afirmar que o Sec-SD representa uma solução para descoberta segura de serviços adequada para ambientes com quantidade elevada de nodos.

5.3 Protótipo

Foi desenvolvido um protótipo para o Sec-SD com uso da linguagem de programação C e do protocolo de transporte UDP unicast e multicast. O software conta com os módulos *cliente*, *provedor* e *provedor redundante*, permitindo assim que os dispositivos que o executam atuem nestes três modos simultaneamente.

O provedor Sec-SD escuta na porta UDP 38000 e no recebimento de uma requisição verifica o campo *ID*, que deve conter o valor de identificação do protocolo Sec-SD, e o *OPCODE*. Se o dispositivo é provedor redundante para algum serviço, a probabilidade de resposta P_r é verificada dependendo do tipo de mensagem (ex. requisições de autenticação podem ou não ser processadas, enquanto que as solicitações de chave de grupo sempre são processadas).

Em caso de a mensagem ter de ser processada, esta é enviada para o bloco funcional específico (ex. analisador de requisições de autenticação, analisador de requisições de descoberta, etc.). Os provedores não redundantes processam todas as requisições. Para *timeout* no envio das mensagens, foi aplicado o intervalo de dois segundos.

No lado do cliente, as mensagens recebidas tem seu *ID* e *OPCODE* verificados. Neste caso, o *OPCODE* permite também que ele verifique se a mensagem é ou não esperada (por exemplo, se nenhuma solicitação de chave de grupo foi enviada, uma chave que seja recebida deve ser descartada). Após essa verificação inicial, a mensagem é enviada para o bloco funcional correspondente, a fim de ser processada.

A seção a seguir mostra uma avaliação do desempenho obtido com o uso do protótipo do Sec-SD no contexto do sistema *Local Peer-to-Peer Protocol* (LP2P).

5.4 Caso de uso: Descoberta de serviços em um sistema Peer-to-Peer projetado para ambientes LAN

5.4.1 O projeto LP2P

O Local Peer-to-Peer-Protocol (LP2P) propõe um sistema de compartilhamento peer-to-peer para uso em redes locais, empregando uma abordagem descentralizada e escalável. O projeto LP2P combina as características específicas das redes LAN, como a alta taxa de transmissão e a baixa latência de comunicação com os aspectos comumente encontrados nas redes P2P, como por exemplo os métodos utilizados para busca de conteúdos e formação de base de conhecimento.

O LP2P está em desenvolvimento no grupo de Redes de Computadores e Sistemas Distribuí-

dos do PIPCA e prevê módulos para gerenciamento de conteúdo compartilhado, comunicação aberta e segura, fatores de replicação (que possibilitam ajuste fino para controle do consumo de disco nas estações), integração com clientes que atuam como *front-end* entre o sistema e o usuário.

Para obter melhor aproveitamento da largura de banda, as mensagens de controle do protocolo LP2P (como por exemplo, adição e remoção de conteúdos) são enviadas em multicast. Devido ao fato de o LP2P ter como escopo de funcionamento a rede local, as questões relacionadas a escalabilidade da comunicação multicast não representam uma preocupação.

Levando em consideração os aspectos citados anteriormente, a arquitetura proposta apresenta diversas vantagens quando comparada ao modelo tradicional cliente-servidor e também a outras soluções que empregam a topologia peer-to-peer mas que utilizam algum gerenciamento centralizado. Desta forma, o LP2P pode ser utilizado em universidades e empresas como alternativa aos servidores dedicados na rede em caso de falhas, ou ainda, substituindo completamente o armazenamento centralizado, minimizando os problemas de desempenho e de tolerância a falhas e reduzindo também os custos com equipamentos e pessoal qualificado para gerenciamento.

5.4.2 Requisitos do sistema LP2P quanto a descoberta de serviços

A seguir são destacados alguns dos principais requisitos para a descoberta dos serviços no sistema LP2P:

- A descoberta de serviços deve poder ser feita com ou sem o uso de mecanismos de segurança;
- Os conteúdos que não são confidenciais devem ser anunciados para todos os usuários no momento em que se tornam disponíveis na rede e todas as requisições para os mesmos devem ser respondidas;
- As informações relacionadas aos compartilhamentos restritos não devem ser anunciadas na rede, ao contrário do que acontece com os conteúdos abertos. A configuração do tipo de serviço (aberto ou restrito) deve ser feita pelo proprietário do compartilhamento, que pode tomar como base para as suas decisões as características do ambiente. Por exemplo, o LP2P dentro de um escritório e em uma conferência exigem níveis de segurança diferenciados – no primeiro caso, o acesso a determinados compartilhamentos poderá ser liberado a todos os usuários, enquanto que no segundo, o proprietário pode definir que nenhum compartilhamento seu será divulgado (e consequentemente acessado) sem que o usuário forneça as suas credenciais;
- Os serviços confidenciais não devem ser anunciados espontaneamente: ao invés disso, os provedores devem aguardar o recebimento de requisições e verificar se o cliente possui

credenciais para acessar o serviço;

Entre as características desejáveis na descoberta de serviços do LP2P, está a facilidade de uso do sistema que inclui, entre outros aspectos, a apresentação de informações importantes para a escolha de um ou outro serviço (no caso, os compartilhamentos disponíveis). Por exemplo, é desejável que o usuário saiba os nomes dos compartilhamentos que são disponibilizados por cada participante da rede e não apenas se determinado usuário possui pastas para compartilhar.

Neste ponto, é importante destacar que o Sec-SD permite que a descoberta de serviços seja feita somente de modo seguro. Levando em consideração as necessidades do LP2P e as características identificadas nos protocolos descritos no capítulo 3, verificou-se que o mDNS e o DNS-SD, descritos na seção 3.4 possuem diversos aspectos positivos e desejáveis para o contexto do LP2P, entre os quais podem ser destacados a descoberta por tipos de serviços, o acesso através de nomes semelhantes aos que são utilizados no DNS e o modo de operação sem diretórios.

Por estas razões, com o objetivo de atender aos requisitos do sistema LP2P, optou-se pela utilização do Sec-SD e do mDNS/DNS-SD para descoberta e anúncio de serviços do LP2P.

Para atingir os objetivos descritos, optou-se por utilizar a ferramenta Avahi[1]. O Avahi foi implementado em linguagem C e é uma solução baseada em mDNS/DNS-SD de código aberto para sistemas Linux.

5.4.3 Os serviços LP2P

O LP2P prevê dois tipos de serviços distintos para designar os compartilhamentos abertos e os compartilhamentos restritos — o open-lp2p e o restricted-lp2p, respectivamente. O uso do Sec-SD e do mDNS/DNS-SD se dará da seguinte forma: na descoberta dos serviços open-lp2p, será utilizado o mDNS/DNS-SD, enquanto que na descoberta de serviços restricted-lp2p, será utilizado o Sec-SD.

Este controle é realizado pelo cliente Sec-SD/Avahi, que repassa as informações correspondentes a cada tipo de serviço, para anúncio ou decoberta, de acordo com cada caso. O cliente foi implementado em linguagem C e testado no sistema Ubuntu 9.10. Sua interação com o Avahi e o Sec-SD é mostrada na Figura 26.

A seguir são descritos os procedimentos para descoberta de cada tipo de serviço.

5.4.3.1 Descoberta e anúncio do serviço open-lp2p

Este serviço é oferecido por peers que possuem compartilhamentos abertos a todos os usuários da rede LP2P. Desta forma, um peer que possui pelo menos um compartilhamento aberto ingressa na rede e anuncia em multicast o serviço open-lp2p. A máquina também deve responder as eventuais requisições para este tipo de serviço.

[1] http://avahi.org

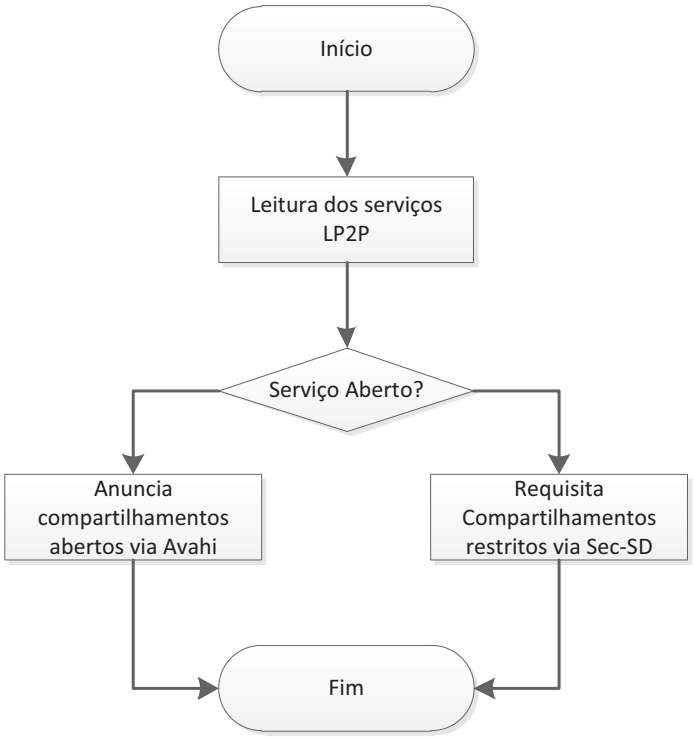

Figura 26: Cliente Sec-SD/Avahi.

Confome foi explicado no capítulo 3, seção 3.4, o DNS-SD possibilita a descoberta por tipos (no caso, open-lp2p) e cada serviço é identificado individualmente por um nome (ex: usuario@usuário-desktop). O endereço e a porta do serviço também são informados sob a forma de endereço IP e nome como os do DNS (usuário-dektop.local). No registro TXT é possível a inserção de texto livre, com informações adicionais do serviço. No caso do LP2P, o TXT dos anúncios contém os nomes dos compartilhamentos públicos (ex: Artigos, Material_PIPCA, Fotos) e a versão do protocolo.

Se um peer possuir alguns compartilhamentos abertos e outros de acesso restrito, ele responderá normalmente as requisições pelo tipo open-lp2p e informará os seus compartilhamentos abertos. Por outro lado, se ele possuir apenas compartilhamentos restritos, deverá permanecer em silêncio diante das requisições por conteúdos abertos. O anúncio dos serviços open-lp2p se dá pela interação do cliente Sec-SD/Avahi com o módulo *avahi-client*[2].

5.4.3.2 Descoberta do serviço restricted-lp2p

O serviço restricted-lp2p é oferecido por peers que tem compartilhamentos de acesso restrito. Estas máquinas podem ou não disponibilizar compartilhamentos open-lp2p, que são anunciados seguindo os procedimentos descritos anteriormente. Os compartilhamentos restricted-lp2p não são anunciados quando se tornam disponíveis na rede. Ao invés disso,

[2]http://avahi.org/wiki/ArchitecturalOverview

o *daemon* do Sec-SD que é executado nos peers deve aguardar por requisições e verificar as credenciais dos potenciais clientes.

Um usuário do LP2P que cria um compartilhamento seguro atribui a este um nome (o identificador do serviço) e uma senha, tornando-se o primeiro participante do grupo que tem acesso ao compartilhamento. Ao liberar o acesso para outros usuários da rede LP2P, ele divulga para estes o nome do compartilhamento e a senha, que servirão para que um membro do grupo possa descobrir outros membros na rede, uma vez que compartilhamentos restritos não são anunciados de forma espontânea. Para obter a chave de grupo, os nodos devem realizar os procedimentos para autenticação via Sec-SD descritos no capítulo 4.

De posse da chave, o cliente pode visualizar o conteúdo do compartilhamento através do envio da mensagem *Descoberta*. O acesso aos compartilhamentos (ex: cópia de arquivos) é feito através do protocolo LP2P. Para isso, o cliente deve selecionar o arquivo desejado e o estabelecer chaves de seção junto ao protocolo LP2P (ROCHA, 2011).

Duas máquinas que possuem acesso a um compartilhamento restricted-lp2p são provedores redundantes do serviço designado pelo identificador atribuído ao compartilhamento. Neste ponto é válido reforçar que a busca do Sec-SD usa como critério o identificador e a senha do serviço (a partir dos quais são gerados os hashes enviados nas requisições), ao invés dos tipos empregados no DNS-SD.

Assim, cada nodo da rede que possui acesso a um determinado compartilhamento restricted-lp2p ingressa na rede e requisita o serviço correspondente, se não houver nenhum outro provedor ativo ele mesmo deve gerar a chave de grupo e aguardar por requisições de outros provedores. Os serviços restricted-lp2p representam um caso onde todos os os nodos que tem as credenciais para descoberta são provedores redundantes (ou seja, ao receberem a chave de grupo podem também responder requisições).

Para o cálculo do coeficiente de redundância C_r, uma máquina que oferece somente serviços restricted-lp2p para descoberta via Sec-SD (podendo também oferecer diversos serviços abertos) possui 100% de redundância, ou seja possui coeficiente de redundância $C_r = S$ e probabilidade de resposta $P_r = 0.5$.

A seção a seguir descreve aspectos de desempenho do protótipo desenvolvido.

5.4.4 Desempenho do protótipo do Sec-SD

A requisição de autenticação enviada pelo cliente quando este entra na rede contém um hash do identificador do serviço concatenado com a senha do mesmo e com um time-stamp T_{s1}, também enviado em claro na mensagem. Quando recebe uma requisição, o provedor de serviços verifica a mesma gerando um hash do identificador de cada um dos seus serviços concatenado com a respectiva senha e com o time-stamp recebido e compara com com o hash extraído da requisição.

O tempo médio de processamento de uma requisição de autenticação, T_p é proporcional

ao tempo gasto no cálculo de um hash e na comparação de duas *strings* (respectivamente, o hash recebido e o hash gerado). Este processo deve ser repetido para cada um dos serviços disponibilizados pelo provedor até que seja encontrada uma correspondência, ou no pior caso, para todos os serviços.

Com o objetivo de estimar este tempo, optou-se por executar repetidamente o bloco funcional do protótipo do Sec-SD que é responsável pela análise das requisições de autenticação (onde são gerados os hashes MD5 de 128 bits com uso da biblioteca *Crypt* do UNIX e é feita a comparação com o hash recebido).

O experimento descrito acima, assim como os demais relatados neste trabalho, foram realizados em uma plataforma Intel Core 2 Duo com clock de 3GHZ e 4GB de memória executando o sistema operacional Ubuntu 9.10 no laboratório do PIPCA - UNISINOS, com um switch 3COM 10/100/1000.

Foi observado então que o tempo médio t_p transcorrido na execução da função descrita acima foi de 1 ms. Assim, T_p é dado por:

$$T_p = S.t_p$$

Onde S é a quantidade de serviços oferecidos por um dado dispositivo.

O tempo total necessário para um cliente obter a chave de grupo de um serviço compreende os tempos para a geração e envio da requisição, processamento da requisição nos provedores, envio da resposta, solicitação da chave de grupo, envio da chave de grupo criptografada e processamento da chave no cliente (descriptografia). As mensagens de solicitação e envio da chave podem exigir a execução de algoritmos de assinatura digital e de criptografia assimétrica no nivel 2 de segurança do Sec-SD.

Para criptografia simétrica foi utilizado o algoritmo AES (NIST, 1997) com tamanho de chave e de bloco de 256 bits. Para criptografia assimétrica foi usado o algoritmo RSA (JONSSON; KALISKI, 2003), através do software GPG (*GNU Privacy Guard*)[3].

A Tabela 4 mostra os tempos médios gastos com a execução das tarefas descritas acima. Para obtenção destes dados o cliente Sec-SD/Avahi foi executado 30 vezes fazendo solicitações de serviços restricted-lp2p. Foram descobertos serviços com uso do primeiro e do segundo nível de segurança do Sec-SD (15 vezes para cada nível).

Os dados para autenticação foram buscados em um arquivo com informações sobre serviços (fictícios) do LP2P. Para fins de comparação, foi efetuada a descoberta de serviços exclusivos e redundantes. No caso dos serviços redundantes foi aplicado um artifício (com $P_r = 0$) para que a primeira requisição sempre falhasse e a segunda fosse respondida, conforme a especificação. O provedor processou 100 serviços durante a análise das requisições de autenticação (o serviço requisitado estava na última posição da listagem).

Como os provedores do serviço restricted-lp2p atuam sempre em redundância (as máquinas

[3] www.gnupg.org

clientes tornam-se provedoras após receberem a chave de grupo), o comportamento esperado na descoberta de serviços exclusivos foi reproduzido neste experimento através do emprego de um provedor que gera a chave de grupo e outro que ingressa na rede e busca pelo seu serviço para obter esta chave. Ambos os provedores possuem 100 serviços e $C_r = 1$, o que resulta em uma probabilidade de resposta $P_r = 0.99$. Assim, o resultado é similar ao que seria obtido com um provedor exclusivo, fora do ambiente LP2P.

Tabela 4: Tempos para descoberta de serviços

Tarefa	**Sistema**	**Tempo (s)**
Descoberta do serviço open-lp2p	Avahi	0.012
Descoberta de um serviço restricted-lp2p	Sec-SD (nível 1)	0.059
Descoberta de um serviço restricted-lp2p	Sec-SD (nível 2)	0.078
Descoberta de um serviço restricted-lp2p	Sec-SD (nível 1 com timeout)	2.063
Descoberta de um serviço restricted-lp2p	Sec-SD (nível 2 com timeout)	2.089

O cliente Sec-SD/Avahi também foi usado para descobrir um serviço open-lp2p disponibilizado por uma máquina no ambiente de testes. Para isso, a ferramenta *avahi-browse* foi executada também 15 vezes. Este tempo foi medido com o intuito de permitir a comparação do protótipo do Sec-SD com uma ferramenta que utiliza uma abordagem distribuída como é o caso do mDNS/DNS-SD, onde qualquer máquina pode atuar como cliente e provedor (tal como ocorre no sec-SD), porém sem o uso de mecanismos de segurança. Os tempos obtidos são mostrados na Tabela 4.

Observou-se que para obter uma chave de grupo utilizando o nível de segurança 1 do Sec-SD (com uso de criptografia simétrica), o tempo transcorrido é cerca de 5 vezes o tempo necessário para a descoberta de um serviço com o Avahi. Com o uso de criptografia assimétrica, o Sec-SD necessita de 0.066 segundos a mais do que o Avahi para descobrir um serviço.

As duas situações onde o sistema foi forçado a atingir o *timeout* (de dois segundos neste protótipo) representam o caso em que um cliente de um serviço provido em redundância fica sem resposta devido ao cálculo de P_r resultar em um valor baixo devido ao fato de o provedor ser redundante para diversos serviços ou ainda, por existir somente um provedor ativo na rede no momento em que é feita a requisição. Porém, tanto com uso de criptografia simétrica quanto assimétrica os clientes obtiveram resposta em menos de 2.1 segundos.

Considerando os mecanismos de segurança utilizados pelo Sec-SD durante o processo de descoberta de serviços, que compreende a negociação da chave de grupo e o envio de mensagens de descoberta, os tempos podem ser considerados satisfatórios, superando 0.1 segundo por serviço somente nos casos em que ocorre timeout.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado o Sec-SD (*Secure Service Discovery Protocol*), um protocolo para descoberta segura de serviços voltado aos ambientes LAN. Dentre suas características, estão o processo reativo (VERVERIDIS; POLYZOS, 2008) de relacionamento das informações aos serviços presentes na rede aliado a sua arquitetura descentralizada. O Sec-SD também trata a redundância de provedores para um serviço de forma que o usuário visualize somente um serviço ativo na rede, podendo exibir informações adicionais sobre o mesmo de acordo com as preferências de seu proprietário (ex: administrador da rede).

O desenvolvimento do Sec-SD foi motivado pelo interesse crescente da comunidade acadêmica e da indústria por aplicações destinadas a descoberta automática de recursos disponíveis nas redes de computadores, onde surge de forma natural a necessidade de tratar questões relacionadas a segurança. Neste contexto, as necessidades variam de acordo com o ambiente, mas de modo geral, as tecnologias seguras para descoberta de serviços devem evitar o anúncio de falsos serviços e/ou sua aceitação pelos demais nodos da rede e possibilitar que as partes envolvidas neste processo disponham de mecanismos para verificar se a entidade que está no outro extremo da comunicação realmente é quem ela afirma ser.

Neste ponto, é importante destacar que o tráfego de mensagens em protocolos seguros tende a ser maior em razão do envio de credenciais para autenticação, certificados, etc., o que também representa uma questão a ser tratada no projeto de tais protocolos.

Dentre as tecnologias estudadas para embasar este trabalho, foram encontradas diversas soluções para prover as características citadas acima. Porém, as tarefas relacionadas a autenticação de clientes e provedores de serviço são desempenhadas por servidores centralizados. Além disso, nenhuma das soluções encontradas na literatura trata a questão da redundância de provedores.

Assim, o Sec-SD traz como contribuições a descentralização da autenticação de usuários, que é realizada pelos provedores de serviço e possibilita a obtenção de uma chave que é utilizada na obtenção de informações sobre os serviços, o tratamento de provedores redundantes e a supressão do tráfego de respostas repetidas dos mesmos. Aliado a isso, o Sec-SD facilita a formação de *logs* nos nodos da rede através do formato utilizado nas mensagens de descoberta, tendo em vista permitir o gerenciamento adequado das atividades dos clientes e provedores, motivado pelo fato desta característica ser encontrada em diversos mecanismos de autenticação utilizados na atualidade.

No que diz respeito a facilidade de uso, o Sec-SD utiliza identificadores e senhas para os serviços, que representam um mecanismo amigável para os usuários humanos, utilizando estas credenciais para a obtenção de uma chave (que é utilizada na interação com os provedores de serviços) no momento em que os nodos ingressam na rede. Para permitir que os serviços disponíveis na rede sejam facilmente identificados pelos usuários é utilizado um campo que pode conter um texto configurado pelos proprietários do serviços, onde podem constar infor-

mações necessárias para que ele escolha os serviços que melhor atendam as suas necessidades.

O Sec-SD foi avaliado em relação ao tráfego de mensagens do tipo resposta de autenticação em cenários com quantidade de nodos e de serviços distintos, tendo apresentado desempenho dentro dos valores aceitáveis (91 KBytes por minuto no pior caso, em uma rede de 100 Mb/s).

O protótipo desenvolvido foi avaliado no contexto da aplicação LP2P (*Local Peer-to-Peer Protocol*), que é um sistema para compartilhamento de arquivos para redes LAN. Neste cenário, o Sec-SD teve seus tempos de descoberta comparados aos do Avahi, que é uma ferramenta que utiliza mDNS/DNS-SD, que são tecnologias cuja abordagem é distribuída, tal como ocorre com o sec-SD, porém sem o uso de mecanismos de segurança. Durante os testes, o Sec-SD apresentou tempos de descoberta inferiores a 2.1 segundos nos casos onde ocorreu timeout na primeita tentativa e tempos menores que 0.1 segundo nos casos em que se obteve resposta já na primeira requisição de descoberta.

Com estes resultados, é possível afirmar que o Sec-SD representa uma solução para descoberta de serviços segura que pode ser utilizada em ambientes variados, tais como escritórios, redes domésticas, empresas e universidades.

Como trabalho futuro, pode ser desenvolvido um mecanismo para notificar clientes sobre a desativação de serviços oferecidos por um único provedor , visto que na versão atual o Sec-SD oferece ao administrador da rede a possibilidade de configurar em cada máquina os intervalos para envio da mensagem *Descoberta*, que tem como objetivos permitir que o cliente receba informações relacionadas a um serviço e que de tempos em tempos ele verifique se este serviço continua disponível.

Ainda como trabalho futuro, pode ser realizada a integração do Sec-SD aos protocolos mDNS/DNS-SD, que constituem uma solução totalmente descentralizada para descoberta de serviços e autoconfiguração de endereços largamente utilizada, mas não dispõe de nenhum mecanismo de segurança. Desta forma, o Sec-SD pode prover confidencialidade, autenticação e integridade na descoberta de serviços e ao mesmo tempo preservar o tipo de arquitetura utilizado por estas tecnologias. As requisições mDNS/DNS-SD podem encapsular as requisições Sec-SD para serviços seguros, sendo mantido o formato tradicional das requisições para serviços públicos.

REFERÊNCIAS

APPLE. **Bonjour Overview**. Informação técnica sobre o sistema Bonjour. Disponível em: <http://developer.apple.com/mac/library/documentation/cocoa/Conceptual/NetServices/Articles/about.html>. Acesso em: junho 2010.

ARENDS, R.; AUSTEIN, R.; LARSON, M.; MASSEY, D.; ROSE, S. **RFC 4035**: protocol modifications for the dns security extensions. Internet Engineering Task Force (IETF).

AUBINEAU, Y. **Towards Seamless Local Networking**. 2004. Masther Thesis — Oxford Brookes University, 2004.

BRODER, A.; MITZENMACHER, M. Network Applications of Bloom Filters: a survey. **Internet Mathematics**, [S.l.], v. 1, n. 4, p. 485–509, 2002.

CAESAR, M.; CASTRO, M.; NIGHTINGALE, E. B. Virtual Ring Routing: network routing inspired by DHTs. In: IN PROC. OF ACM SIGCOMM, 2006, New York, NY, USA. **Anais...** [S.l.: s.n.], 2006. p. 351–362.

CHESHIRE, S.; KROCHMAL, M. **Multicast DNS**. Internet Draft, Internet Engineering Task Force (IETF). http://tools.ietf.org/html/draft-cheshire-dnsext-multicastdns-14.

CHESHIRE, S.; KROCHMAL, M. **DNS-Based Service Discovery**. Internet Draft, Internet Engineering Task Force (IETF). http://tools.ietf.org/html/draft-cheshire-dnsext-dns-sd-10.

O'REILLY (Ed.). **Zero Configuration Networking**: the definitive guide. Sebastopol, CA, USA: O'Reilly, 2006.

CLAUSEN, T.; JACQUET, P. **RFC 3626**: optimized link state routing protocol (olsr). Internet Engineering Task Force (IETF).

COHEN, J.; AGGARWAL, S. **General Event Notification Architecture Base**. Internet Draft, Internet Engineering Task Force (IETF). http://tools.ietf.org/html/draft-cohen-gena-p-base-01.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Distributed Systems**: concepts and design. 4th. ed. Boston, MA, USA: Addison Wesley, 2005.

D. CARREL, L. G. **The TACACS+ Protocol Version 1.78**. [S.l.: s.n.], 1997. Internet Draft: Internet Engineering Task Force (IETF).

DIERKS, T.; RESCORLA, E. **RFC 5246**: the TLS protocol version 1.2. Internet Engineering Task Force (IETF).

EDWARDS, W. Discovery Systems in Ubiquitous Computing. **IEEE Pervasive Computing**, Piscataway, NJ, USA, p. 70–77, Apr. 2006.

ELLISON, C. **UPnP Security Ceremonies**. UPnP Forum. http://www.upnp.org/specs/sec/UPnP-sec-UPnPSecurityCeremonies-v1.pdf.

ENGELSTAD, P.; EGELAND, G. The Handbook of Mobile Middleware. In: _____. . Boston, MA, USA: Auerbach Publications, 2006.

FIELDING, R.; GETTYS, J.; MOGUL, J.; FRYSTYK, H.; LEACH, P.; BERNERS-LEE, T. **RFC 2616**: hypertext transfer protocol – HTTP 1.1. Internet Engineering Task Force (IETF).

FOROUZAN, B. **Comunicação de Dados e Redes de Computadores**. 4th. ed. São Paulo, SP, Brasil: McGraw-Hill, 2008.

GOLAND, Y.; CAI, T.; LEACH, P.; GU, Y.; ALBRIGHT, S. **Simple Service Discovery Protocol/1.0**. Internet Draft, Internet Engineering Task Force (IETF). http://tools.ietf.org/html/draft-cai-ssdp-v1-03.

GOLBECK, J. Trust on the World Wide Web: a survey. **Foundations and Trends in Web Science**, [S.l.], v. 1, n. 2, p. 131–197, 2008.

HELAL, S.; DESAI, N.; VERMA, V. Konark – A Service Discovery and Delivery Protocol for Ad Hoc Networks. In: THIRD IEEE CONFERENCE IN WIRELESS COMMUNICATION (WCNC), 2003, New Orleans, LA, USA. **Anais...** [S.l.: s.n.], 2003.

HOUSLEY, R.; FORD, W.; POLK, W.; SOLO, D. **RFC 2459**: internet x.509 public key infrastructure certificate and crl profile. Internet Engineering Task Force (IETF).

IEEE. **IEEE 802.15.1**: wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (wpans). IEEE Standard, IEEE Computer Society.

IEEE COMPUTER SOCIETY. **IEEE 802.11**: wireless lan medium access control (mac) and physical layer (phy) specifications. New York, NY, USA, 2007. Revision of IEEE Std 802.11-1999.

GUTTMAN, E.; PERKINS, C.; VEIZADES, J.; DAY, M. **RFC 2608**: service location protocol version 2. [S.l.: s.n.], 1999. Internet Engineering Task Force (IETF).

JONSSON, J.; KALISKI, B. **RFC 3447**: public-key cryptography standards (pkcs) 1: rsa cryptography specifications version 2.1. Internet Engineering Task Force (IETF).

JUSZCZYK; LUKASZ; DUSTDAR; SCHAHRAM. A middleware for service oriented communication in mobile disaster response environments. In: MPAC 08: PROCEEDINGS OF THE 6TH INTERNATIONAL WORKSHOP ON MIDDLEWARE FOR PERVASIVE AND AD-HOC COMPUTING, 2008, New York, NY, USA. **Anais...** ACM, 2008. p. 37–42.

KREBS, M.; KREMPELS, K.-H.; KUCAY, M. Service Discovery in Wireless Mesh Networks. In: IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE (WCNC) 2008, 2008, Las Vegas, NV, USA. **Anais...** [S.l.: s.n.], 2008.

LINN, J. **RFC 1964**: the kerberos version 5 gss-api mechanism. Internet Engineering Task Force (IETF).

METTÄLÄ, R.; BISDIKIAN, C.; BOUET, S.; INOUYE, J.; MILLER, B.; MORLEY, K.; MULLER, T.; ROTTER, M.; SLOTBOOM, E. **Bluetooth Protocol Architecture**. White Paper. http://www.bluetooth.com/English/Technology/Building/Pages/ResearchandWhitePapersDetail.aspx?ID=14.

MOCKAPETRIS, P. **RFC 1035**: domain names - implementation and specification. Internet Engineering Task Force (IETF).

NIST. **Advanced Encryption Standard (AES)**. National Institute of Standards and Technology, (FIPS PUB 197). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

NIST. **Data Encryption Standard (DES)**. National Institute of Standards and Technology, (FIPS PUB 46-3). http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

PRESSER, A.; FARRELL, L.; KEMP, D.; LUPTON, W.; S.TSURUYAMA; S.ALBRIGHT. **UPnP Device Architecture 1.1**. UPnP Forum. www.upnp.cn/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf.

RESCORLA, E. **RFC 2631**: diffie-hellman key agreement method. Internet Engineering Task Force (IETF).

RESCORLA, E. **RFC 2818**: HTTP over TLS. Internet Engineering Task Force (IETF).

RIGNEY, C. **RFC 2866**: RADIUS accounting. [S.l.: s.n.], 2000. Internet Engineering Task Force (IETF).

RIGNEY, C.; WILLENS, S.; RUBENS, A.; SIMPSON, W. **RFC 2865**: remote authentication dial in user service (RADIUS). [S.l.: s.n.], 2000. Internet Engineering Task Force (IETF).

ROCHA, É. S. **LP2P**: local peer-to-peer protocol. 2011. Masther Thesis — Universidade do Vale do Rio dos Sinos, 2011.

ROCHA, É. S.; MARCON, D. S.; ÁVILA, R. B. Comunicação Peer-to-Peer aplicado a Redes Locais. In: ESCOLA REGIONAL DE REDES DE COMPUTADORES, ERRC, 8., 2010, Alegrete, Brazil. **Anais...** [S.l.: s.n.], 2010.

SAIRAM, K.; GUNASEKARAM, N.; REDDY, S. Bluetooth in Wireless Communication. **IEEE Communications Magazine**, [S.l.], p. 90–96, June 2002.

SCHNEIER, B. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). In: CAMBRIDGE SECURITY WORKSHOP PROCEEDINGS 1993, 1993, Cambridge, UK. **Anais...** [S.l.: s.n.], 1993.

SCHNEIER, B. **Applied Cryptography**. 2th. ed. New York, NY, USA: John Wiley & Sons, 1995.

SNELL, J.; TIDWELL, D.; KULCHENKO, P. **Programming Web services with SOAP**. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2002.

STALLINGS, W. **Criptografia e Segurança de Redes**. 4th. ed. São Paulo, SP, Brasil: Pearson Prentice Hall, 2008.

TANENBAUM, A. **Redes de Computadores**. 4th. ed. Rio de Janeiro, RJ, Brasil: Campus, 2003.

THOMSEN, B.; BAKER, T.; BLAKE-WILSON, S.; WILLEY, D.; XYDIS, T.; GEHRMANN, C. **Bluetooth Security**. [S.l.]: Bluetooth SIG Security Expert Group, 2002. White Paper. http://www.bluetooth.com/English/Technology/Building/Pages/ResearchandWhitePapersDetail.aspx?ID=18.

VERVERIDIS, C.; POLYZOS, G. Service Discovery for Mobile Ad Hoc Networks: a survey of issues and techniques. **IEEE Communications Surveys & Tutorials**, [S.l.], v. 10, n. 3, p. 30–45, 2008.

ZHU, F.; MUTKA, M. Service Discovery in Pervasive Computing Environments. **IEEE Pervasive Computing**, Piscataway, NJ, USA, p. 81–90, 2005.

ZHU, F.; MUTKA, M.; BIVALKAR, A.; DEMIR, A.; LU, Y.; CHIDAMBARM, C. Toward Secure and Private Service Discovery Anywhere Anytime. **Frontiers of Computer Science in China**, [S.l.], v. 4, p. 311–323, 2010.

ZHU, F.; MUTKA, M.; NI, L. A Private, Secure, and User-Centric Information Exposure Model for Service Discovery Protocols. In: IEEE TRANSACTIONS ON MOBILE COMPUTING, 2006, Piscataway, NJ, USA. **Anais...** [S.l.: s.n.], 2006. v. 5, n. 4.

ZHU, F.; ZHU, W.; MUTKA, M.; NI, L. Private and Secure Service Discovery via Progressive and Probabilistic Exposure. **IEEE Transactions on Parallel and Distributed Systems**, Piscataway, NJ, USA, Nov. 2007.